

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY  
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

**ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY**

DEPARTMENT OF CONTROL AND INSTRUMENTATION

**ŘÍZENÍ SCARA ROBOTU POMOCÍ ŘÍDICÍHO SYSTÉMU  
BECKHOFF**

CONTROL OF THE SCARA ROBOT USING THE BECKHOFF CONTROL SYSTEM

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**Adam Madzia**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. Tomáš Beneš**

**BRNO 2020**

# Bakalářská práce

bakalářský studijní program **Automatizační a měřicí technika**

Ústav automatizace a měřicí techniky

**Student:** Adam Madzia

**ID:** 195383

**Ročník:** 3

**Akademický rok:** 2019/20

**NÁZEV TÉMATU:**

## Řízení SCARA robotu pomocí řídicího systému Beckhoff

### POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je implementace řízení robotu SCARA pomocí měničů a řídicího systému Beckhoff. Součástí práce je vytvořit základní funkcionality pro pohyb robotu.

1. Seznámení se s robotickými systémy a s rozhraními pro jejich řízení.
2. Seznámení se s řídicím systémem Beckhoff.
3. Elektronické schéma zapojení řídicího systému robotu.
4. Realizace elektronického zapojení.
5. Návrh a implementace základních funkcí pro pohyb a řízení robotu včetně komunikačního rozhraní.
6. Řešení otestujte a diskutujte dosažené výsledky.

### DOPORUČENÁ LITERATURA:

JAZAR, Reza N., c2010. Theory of applied robotics: kinematics, dynamics, and control. 2nd ed. New York: Springer. ISBN 14-419-1749-7.

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 8.6.2020

**Vedoucí práce:** Ing. Tomáš Beneš

**doc. Ing. Václav Jirsík, CSc.**  
předseda rady studijního programu

### UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Tato práce se zabývá řízením manipulátoru typu SCARA. V teoretické části práce jsou vysvětleny jednotlivé robotické systémy. Je zde popsána konstrukce robota z hlediska kinematických dvojic a struktur. Dále je popsán konkrétní manipulátor H554BN od firmy Epson a řídicí systém od firmy Beckhoff včetně použitých zařízení. V praktické části je popis realizace zapojení manipulátoru a schéma jeho zapojení. Na konci práce se nachází základní funkce pro ovládání pohonů manipulátoru a funkce pro kinematickou transformaci.

## KLÍČOVÁ SLOVA

Robotické systémy, manipulátor, kinematické struktury, SCARA, H554BN, řídicí systém, Beckhoff, kinematická transformace

## ABSTRACT

This thesis deals with control of the SCARA manipulator. In the theoretical part of the thesis there are explained robotic systems. The construction of a robot is described in terms of kinematic pairs and structures. Next chapter is about the manipulation robot Epson H554BN and Beckhoff control system with used equipment. In the practical part there is a description about the realization of the manipulator connection. Basic functions for controlling manipulator's drives and functions for kinematic transformation are described at the end of the thesis.

## KEYWORDS

Robotic systems, manipulator, kinematic structures, SCARA, H554BN, control system, Beckhoff, kinematic transformation

MADZIA, Adam. *Řízení SCARA robotu pomocí řídicího systému Beckhoff*. Brno, Rok 2020, 73 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce: Ing. Tomáš Beneš



## PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Řízení SCARA robotu pomocí řídicího systému Beckhoff“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno 8. června 2020

.....  
podpis autora

## PODĚKOVÁNÍ

Děkuji vedoucímu mé bakalářské práce panu Ing. Tomášovi Benešovi za odborné rady, doporučení, trpělivost a podnětné návrhy k práci.

Brno 8. června 2020

.....

podpis autora

# Obsah

Úvod	11
<b>1 Robotické systémy a rozhraní pro jejich řízení</b>	<b>12</b>
1.1 Robotické systémy	12
1.1.1 Systémy pro sběr a kontrolu dat	12
1.1.2 Mobilní robotické systémy	13
1.1.3 Manipulační zařízení	13
1.2 Rozhraní pro řízení robotických systémů	14
1.3 Kinematika robotů	14
1.3.1 Typ TTT	15
1.3.2 Typ RTT	15
1.3.3 Typ RRT	17
1.3.4 Typ RRR	17
1.4 Roboty SCARA	18
1.4.1 Stavba robota	19
1.5 Epson H554BN	19
1.5.1 Pohon manipulátoru	21
<b>2 Řídicí systém a komponenty od firmy Beckhoff</b>	<b>23</b>
2.1 Servoměniče	23
2.1.1 Vektorové řízení	23
2.2 Komunikační rozhraní EtherCAT	23
2.3 Coupler a terminály	24
2.4 Průmyslové počítače a operátorské panely	24
<b>3 Elektronické zapojení</b>	<b>27</b>
3.1 Software WSCAD	29
<b>4 Řízení manipulátoru Epson H554BN</b>	<b>31</b>
4.1 Počáteční konfigurace	31
4.2 Komutační offset	32
4.3 Definování a přiřazení proměnných	32
4.4 Funkční bloky pro ovládání motorů	33
4.4.1 MC_Power	33
4.4.2 MC_Reset	34
4.4.3 MC_Stop	34
4.4.4 MC_MoveAbsolute	35

4.4.5	MC_MoveRelative . . . . .	36
4.5	Kinematické řízení - počáteční konfigurace . . . . .	38
4.5.1	Nastavení os . . . . .	38
4.5.2	Nastavení parametrů pro kinematické řízení . . . . .	38
4.6	Implementace kinematického řízení . . . . .	40
4.6.1	Main . . . . .	40
4.6.2	Axes_control_ACS . . . . .	41
4.6.3	Axes_control_MCS . . . . .	41
4.6.4	ACS_Axes_ActualPosition . . . . .	41
4.6.5	Kinematika_transformace . . . . .	41
4.7	Objekty použité v kinematické transformaci . . . . .	42
4.7.1	MC_ReadActualPosition . . . . .	42
4.7.2	ST_KinAxes . . . . .	42
4.7.3	FB_KinConfigGroup . . . . .	43
4.7.4	FB_KinResetGroup . . . . .	44
4.7.5	E_KinStatus . . . . .	45
4.7.6	F_KinGetChnOperationState . . . . .	46
4.7.7	F_KinGetAcsMcsAxisIds . . . . .	46
4.7.8	FB_KinCalcTrafo . . . . .	47
4.7.9	FB_KinUnlockTrafoParam . . . . .	48
4.7.10	FB_KinLockTrafoParam . . . . .	49
	<b>Závěr</b>	<b>51</b>
	<b>Literatura</b>	<b>53</b>
	<b>Seznam symbolů, veličin a zkratek</b>	<b>57</b>
	<b>Seznam příloh</b>	<b>59</b>
	<b>A Schéma zapojení před provedenou opravou</b>	<b>60</b>
	<b>B Schéma zapojení s propojovacím kabelem</b>	<b>67</b>
	<b>C Obsah přiloženého CD</b>	<b>73</b>

# Seznam obrázků

1.1	Členění manipulačních zařízení [2] . . . . .	13
1.2	Kinematická struktura TTT [7] . . . . .	16
1.3	Kinematická struktura RTT [7] . . . . .	16
1.4	Kinematická struktura RRT [7] . . . . .	17
1.5	Kinematická struktura RRR [7] . . . . .	18
1.6	Pracovní prostor SCARA robota [2] . . . . .	19
1.7	Manipulátor Epson H554BN [11] . . . . .	20
1.8	Manipulátor Epson H554BN spolu s příslušenstvím . . . . .	21
1.9	Rozsah pohybu manipulátoru H554BN [11] . . . . .	22
2.1	Panel se zařízeními pro ovládání manipulátoru . . . . .	25
2.2	Blokové schéma zapojení řídicího systému se SCARA robotem . . . . .	26
3.1	Zjednodušené schéma zapojení H554BN [23] . . . . .	28
3.2	Konektor D-SUB CAN 50 . . . . .	29
4.1	Panel pro testování funkčnosti motorů . . . . .	32
4.2	Parametry kinematické skupiny . . . . .	39
4.3	Vývojový diagram pro E_KinStatus [33] . . . . .	45

# Seznam tabulek

1.1	Parametry servomotorů [12]	22
-----	----------------------------	----

## Seznam výpisů

4.1	Funkční blok MC_Power . . . . .	33
4.2	Funkční blok MC_Reset . . . . .	34
4.3	Funkční blok MC_Stop . . . . .	34
4.4	Funkční blok MC_MoveAbsolute . . . . .	35
4.5	Funkční blok MC_MoveRelative . . . . .	37
4.6	Propojení cyklického kanálového rozhraní s I/O kinematického kanálu	40
4.7	Funkční blok MC_ReadActualPosition . . . . .	42
4.8	Struktura ST_KinAxes . . . . .	42
4.9	Funkční blok FB_KinConfigGroup . . . . .	43
4.10	Funkční blok FB_KinResetGroup . . . . .	44
4.11	Funkce F_KinGetChnOperationState . . . . .	46
4.12	Funkce F_KinGetAcsMcsAxisIds . . . . .	46
4.13	Funkční blok FB_KinCalcTrafo . . . . .	47
4.14	Odemknutí a změna parametru kinematické skupiny . . . . .	48
4.15	Uzamčení parametrů kinematické skupiny . . . . .	49

# Úvod

Tato bakalářská práce se bude zabývat řízením SCARA robota pomocí řídicího systému Beckhoff. Součástí této práce bude základní přehled robotických systémů a rozhraní pro jejich řízení. Část práce budu věnovat konstrukci robotů z hlediska kinematiky, kde popíšu základní typy kinematických dvojic a struktur. Budu se věnovat také robotům SCARA, jejich vzniku, k čemu se používají, jaké mají výhody a nevýhody a také přímo jejich konstrukci. Popíši robota, kterého budu řídit, jedná se o staršího robota SCARA H554BN od firmy Epson. Dále se budu zabývat řídicím systémem firmy Beckhoff, od které používám pro ovládání robota jejich hardware a software. Popíšu použité zařízení pro řízení SCARA robota a technologii, kterou mezi sebou komunikují. Jsou zde použity servoměniče AX5201 a AX5203 spolu s couplerem EK1100. K tomuto coupleru jsou připojeny čtyři digitální výstupní terminály EL2008 a dva vstupní digitální terminály EL1124 a koncový kryt EL9011. Vše je ovládané pomocí průmyslového počítače od Beckhoffu C6015-0010. Tento robot byl původně dodáván s vlastním kontrolérem SRC-310, který nebylo možné vyměnit, protože se vázal vždy na konkrétního robota. Z důvodu, že tento kontrolér již není funkční a podpora tohoto typu již skončila, byl nahrazen řídicím systémem od firmy Beckhoff. V rámci práce bude nutné zjistit vyvedení signálů enkodérů pro jednotlivé motory a zapojit je do uvedených servoměničů, ostatní prvky jsou již zapojeny. Je potřeba zjistit, jak je robot včetně jeho příslušenství propojen s řídicím systémem a nakreslit jeho elektronické schéma zapojení. Zapojení budu kreslit v softwaru WSCAD. Po kompletním zapojení robota vytvořím základní funkce pro jeho pohyb. Vytvořím a popíšu funkce pro řízení jednotlivých os samostatně a poté se zaměřím na řízení robota pomocí kinematické transformace. V rámci práce bude možnost absolvování školení ve firmě Beckhoff, kde by se měly ukazovat základy programování v jejich softwaru TwinCAT 3. V tomto softwaru budu také robota programovat.



# 1 Robotické systémy a rozhraní pro jejich řízení

## 1.1 Robotické systémy

Robotické systémy se používají ke zjednodušení výrobních procesů. Může se vhodným systémem nahradit určitý proces ve výrobě, nebo se už v dnešní době dá automatizovat celá výrobní linka, ve které byl předtím nutný lidský zásah. Výroba se díky těmto systémům zrychlí a poklesnou náklady pro vytvoření daného výrobku. Dnes lze nalézt robotické systémy už téměř v každé továrně a tyto systémy se stále rozšiřují a modernizují. [1]

Jako příklad si můžeme uvést montáž, kde se vývoj robotů posouvá velmi rychlým tempem. Průmyslové roboty můžeme nalézt například při sestavování motorových vozidel. Robotická ramena s nástavci různých typů se dnes hojně využívají při svařování, nanášení laku (lidská ruka nemusí nanést lak rovnoměrně na všechna místa), atd. Roboty a robotická ramena se dále využívají na místech, kde nelze vyslat člověka - pracují v různých nebezpečných prostředích, anebo přímo vykonávají pro člověka ohrožující práci - manipulace s chemickými látkami, hašení požárů, odstraňování výbušnin a mnoho dalšího. [2]

### Typy robotických systémů

Robotické systémy můžeme rozdělit na tři typy: [1]

- systémy pro sběr a kontrolu dat;
- mobilní robotické systémy;
- manipulační zařízení.

#### 1.1.1 Systémy pro sběr a kontrolu dat

Slouží pro kontrolu technologického procesu. Většinou se připojují přímo na zařízení, ze kterého data sbírají. Mají za úkol data sbírat, poté je zpracovat a archiovat. Obdržená data slouží například pro kontrolu jakosti, vzdálenou kontrolu, sledování realizace technologického postupu, získání informací o výrobě, získání informací o daném výrobním zařízení, jako je například pracovní vytížení stroje atd. Realizace systémů pro sběr a kontrolu dat probíhá nezávisle na ekonomických informačních systémech. Také je nutno vybrat, která data se mají sbírat a uchovávat. Sběr příliš velkého množství dat může způsobit zahlcení a nepřehlednost. Neměla by se sbírat data, která lze vypočítat z jiných získaných dat. Zároveň by získaná sada

dat měla mít určitou vypovídající hodnotu. Kvalitní informace mohou zlepšit celý proces. [3]

### 1.1.2 Mobilní robotické systémy

Mobilní roboty jsou roboty, které jsou schopny pohybovat se v prostoru. Podle ovládání je můžeme rozdělit na: [4]

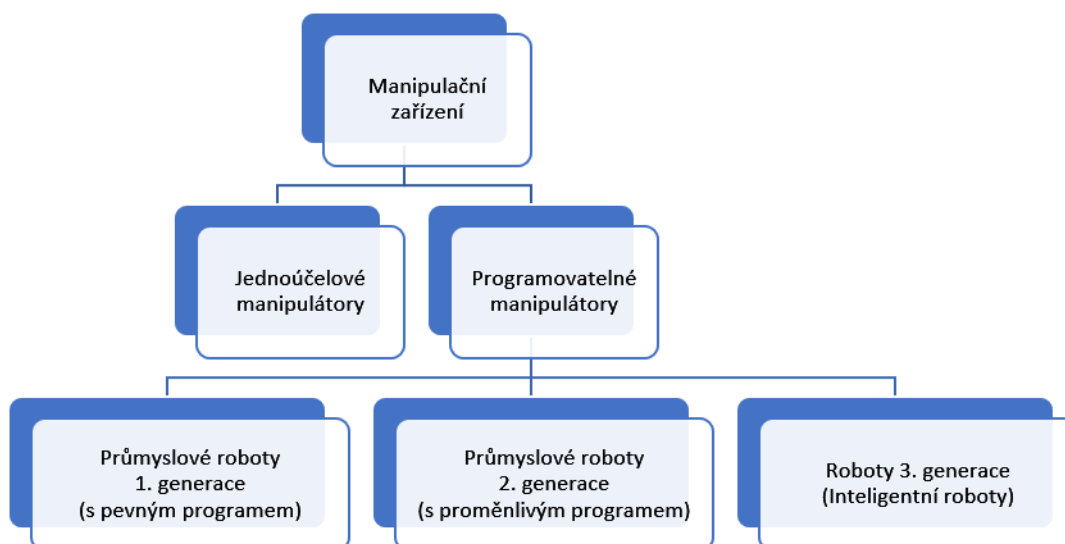
- teleoperované (člověk řídí celého robota dálkově);
- semi-teleoperované (člověk ovládá robota pouze částečně, robot je schopen vykonávat určité procesy samostatně, například při ztrátě komunikace s operátorem se sám přesune na určitý bod);
- autonomní (robot dostane určitý úkol a sám ho vykoná).

### 1.1.3 Manipulační zařízení

Manipulátory a průmyslové roboty nemají jednotnou definici a dělení. Můžeme je rozdělit například podle prováděné práce: [2]

- manipulační (podávání různých předmětů);
- technologické (montážní, lakovací, svařovací,...);
- speciální (práce v radioaktivním prostředí, pod vodou,...);
- univerzální (kombinace různých procesů).

Další dělení může být podle způsobu jejich řízení:



Obr. 1.1: Členění manipulačních zařízení podle způsobu řízení [2]

**Jednouúčelové manipulátory** se používají k automatizaci konkrétního procesu ve výrobě. Jsou limitovány na jednoduché operace potřebné pro dané zařízení, například podavače. [2]

**Roboty první generace** slouží pro opakované vykonávání stejné výrobní činnosti. Tato činnost se skládá z několika pevně daných postupných operací. Je-li nutná jakákoliv změna určité operace nebo celé výrobní činnosti, musí se robot přeprogramovat. [2]

**Roboty druhé generace** obsahují různé typy senzorů nebo jsou vybavené přímo kamerou a jsou schopny přímo kontrolovat správnost daného procesu. [2]

**Roboty třetí generace** jsou schopny samy se učit. Využívají prvků umělé inteligence k přizpůsobení se daným podmínkám a k samostatnému řešení požadovaných úloh. [2]

## 1.2 Rozhraní pro řízení robotických systémů

Robotické systémy mohou být ovládány integrovaným řídicím systémem, který poskytuje přímo výrobce robota, nebo je lze ovládat pomocí programovatelných automatů (PLC). [5]

Pro firmy používající různé druhy zařízení mohou být lepším řešením programovatelné automaty. Pokud by každý robotický systém měl vlastní specializované řízení, bylo by náročnější vytvořit mezi nimi komunikační rozhraní. Jako příklad lze uvést výrobní linku s několika různými roboty a dopravníky, kdy jednotlivé systémy jsou na sobě závislé. Navíc při správě těchto systémů vznikají další náklady spojené s údržbou, nutností znát všechny specializované jazyky výrobců těchto systémů a ovládáním těchto systémů. Programovatelné automaty optimalizují řízení například díky podobnému programování jednotlivých ovládacích prvků a společným softwarovým rozhraním. [5]

Pokud závod již využívá velké množství robotů od jedné firmy nebo požaduje vysoce specializované řízení jistých procesů, může být stále lepší použít speciální řídicí systém výrobců daných robotů. V některých případech použití speciálního řídicího systému výrobce robota je jediné možné řešení, když výrobce neumožňuje implementaci programovatelných automatů. [5]

## 1.3 Kinematika robotů

Roboty se nejčastěji skládají z několika kinematických dvojic, buď rotačních a nebo přímočaře posuvných. Rotační dvojice mohou být v kyvném uložení, nebo uložení

otočném. Translační dvojice mohou být ve smykadlovém uložení, kdy se delší těleso pohybuje v kratším vedení. V opačném případě se jedná o suportové uložení, kdy vedení tvoří delší těleso a po něm se posouvá kratší těleso. Třetí typ posuvné kinematické dvojice je výsuvné (teleskopické) uložení. Translační dvojice se značí velkým písmenem T a rotační dvojice velkým písmenem R. Tyto dvojice lze různě kombinovat a sestavovat, čímž vznikne takzvaná kinematická struktura. [2]

Kinematické struktury mohou být sériové nebo paralelní. Většina manipulátorů a robotů používá sériové řazení. Nevýhodou sériového řazení oproti paralelnímu řazení je menší tuhost (vznik kmitání) a jeho menší přesnost polohování. Všechny vzniklé chyby na jednotlivých kinematických dvojicích se poté projeví na koncovém bodu robota a ovlivní tak jeho pozici. Paralelní struktura má vyšší tuhost a přesnost polohování. Z hlediska konstrukce se může jednat například o tripod nebo až hexapod. Nevýhody těchto struktur jsou vyšší nároky na řízení a při nesprávném naprogramování může dojít ke kolizi jednotlivých vzpěr. [6]

Ze sestavených typů kinematických dvojic se nejvíce používají tyto čtyři struktury: [2]

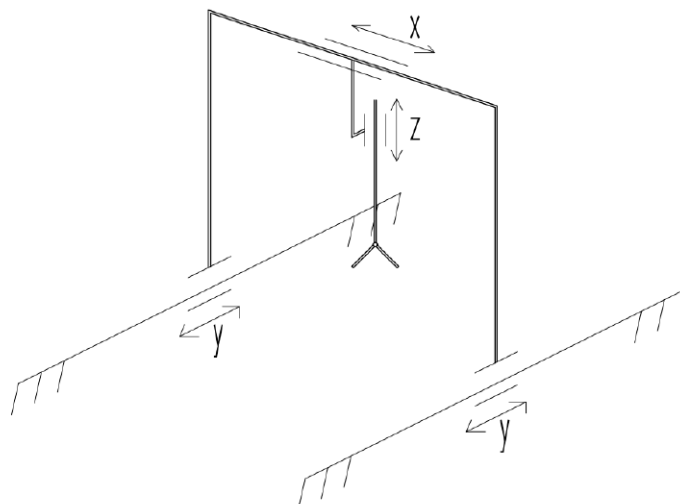
- typ TTT (Tři translační kinematické dvojice);
- typ RTT (Jedna rotační a dvě translační kinematické dvojice);
- typ RRT (Dvě rotační a jedna translační kinematická dvojice);
- typ RRR (Tři rotační dvojice).

### 1.3.1 Typ TTT

Skládá se ze tří posuvných kinematických dvojic. Kinematická struktura přímo používá kartézský souřadný systém. Pracovní hlavice robota se pohybuje v pracovním prostoru ve tvaru krychle nebo kvádrů. Tento typ je z hlediska kinematiky nejpřesnější, má velkou stabilitu a jelikož neobsahuje žádnou rotační kinematickou dvojici, tak nemůže dojít k přetočení přenášeného objektu. Celkový systém je jednoduchý z hlediska řízení. Z důvodu velkého provozního prostoru, který robot zabírá, se tento typ nepoužívá tak často jako ostatní typy manipulátorů. Využívá se hlavně pro větší manipulační prostor, například jako podavač nebo obsluha výrobních strojů. Provedení je znázorněno na obrázku 1.2. [2, 6]

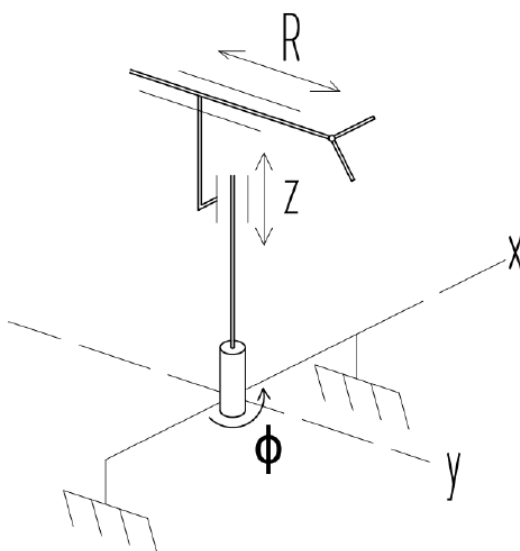
### 1.3.2 Typ RTT

Skládá se z jedné otočné a dvou posuvných kinematických dvojic. Robot potřebuje k operaci s objekty větší manipulační prostor. Pracovní hlavice se pohybuje v pracovním prostoru ve tvaru válcového segmentu. Při manipulaci s objektem dochází



Obr. 1.2: Typ kinematické struktury TTT [7]

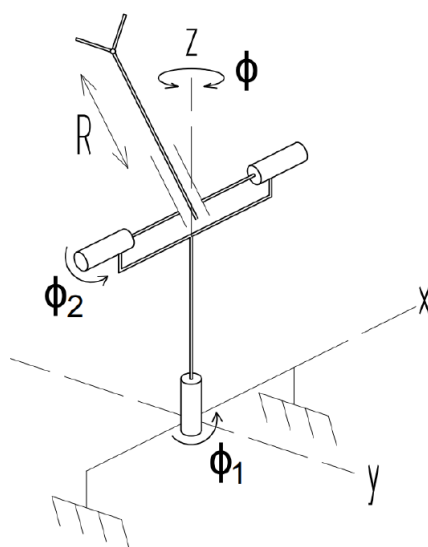
ke změně orientace daného objektu. Jedná se o robustní systém s jednoduchým řízením. Tato struktura se používá například pro práci se vstřikovacími stroji. Provedení je znázorněno na obrázku 1.3. [2, 6]



Obr. 1.3: Typ kinematické struktury RTT [7]

### 1.3.3 Typ RRT

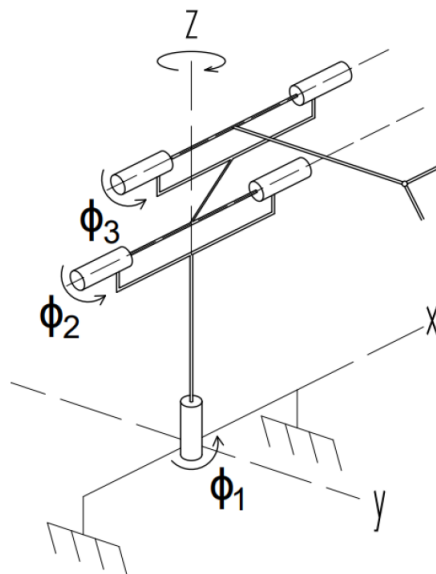
Skládá se ze dvou otočných a jedné posuvné kinematické dvojice. Pracovní hlavice se pohybuje v pracovním prostoru ve tvaru kulového segmentu. Při manipulaci s uchopeným objektem dochází ke změně orientace daného objektu. Tento typ má lépe zvolený manipulační prostor, ale jeho celkový pracovní prostor je menší. Robot má lepší pohyblivost. Z hlediska řízení se jedná o náročnější systém. Je hodně rozšířený u svařovacích linek. Tento typ kinematické struktury poté posloužil jako předloha pro robota typu SCARA. Provedení je znázorněno na obrázku 1.4. [2, 6]



Obr. 1.4: Typ kinematické struktury RRT [7]

### 1.3.4 Typ RRR

Skládá se ze tří otočných kinematických dvojic. Pracovní hlavice operuje v pracovním prostoru ve tvaru tzv. torusu. Vyznačuje se dobrými dynamickými vlastnostmi - operuje ve vysokých rychlostech. Dokáže se vyhýbat různým překážkám díky své dobré manévrovatelnosti. Roboty tohoto druhu zabírají méně místa. Mezi nevýhody patří jeho menší operační dosah, což se dá do značné míry rozšířit díky různým typům pojezdů, na které se může robot namontovat, a jeho náročnější řízení se poté projevuje na menší pracovní přesnosti. Čím delší má robot otočná ramena, tím více klesá jeho opakovatelná přesnost pohybů. Jedná se o nejrozšířenější typ kinematické struktury. Provedení je znázorněno na obrázku 1.5. [2, 6]



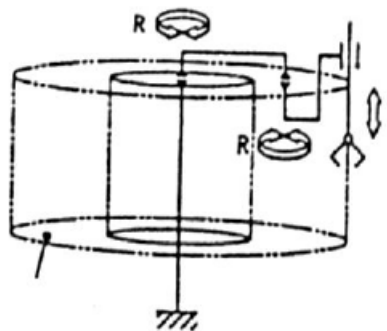
Obr. 1.5: Typ kinematické struktury RRR [7]

## 1.4 Roboty SCARA

Zkratka SCARA znamená Selective Compliant Articulated(nebo Assembly) Robot Arm, do češtiny by se to dalo doslovně přeložit jako „selektivní kompatibilní kloubové robotické rameno“. Prvního SCARA robota navrhl profesor Hiroshi Makino, první prototyp byl poté vytvořen v Japonsku v roce 1978. Mají podobné chování jako lidská ruka, díky tomu se používají k manipulaci s různými výrobky, nakládání nebo k montáži různých dílů. Díky jejich přesnosti se hojně využívají k práci s malými elektronickými součástkami. Jejich cena je o něco menší než cena ostatních typů robotů, a proto se staly oblíbenými v oblasti automatizace montáže. Jsou schopny pracovat v prostředí nebezpečném pro člověka - lze je řídit vzdáleně a zároveň dobře chránit před nebezpečnými látkami, které by mohly poškodit jejich činnost. Mezi jejich hlavní výhody patří jejich rychlost a přesnost, nižší cena oproti ostatním typům, nejsou náročné na velikost provozního prostoru a mají jednodušší ovládání s možností vzdáleného řízení. Mezi jejich nevýhody patří omezené možnosti pohybu pracovní hlavičky - robot nedokáže sám součástku překloupit, ale pouze otočit. Jsou vhodné spíše pro práci s menšími výrobky a dokáží pracovat se součástmi pouze do určité hmotnosti. Většinou se jedná o maximální povolené hmotnosti v řádu jednotek nebo desítek kilogramů. [8, 9]

### 1.4.1 Stavba robota

Roboty SCARA vznikly odvozením z kinematické struktury typu RRT. Pracovní prostor pracovní hlavice je zobrazen na obrázku 1.6. Jedná se o válcový prstenec. [2] Ve většině případů má robot čtyři osy a čtyři stupně volnosti. Zpravidla je první a druhá kinematická dvojice rotační a obě se otáčí podél osy Z. Někdy může být druhá dvojice translační a pohybovat se po ose Z. Číslování os začíná od strany základny, ke které je robot uchycen. Pracovní hlavice je umístěna na translační kinematické dvojici, která posouvá hlavici nahoru a dolů po ose Z a je zároveň schopná se podél ní otáčet. Díky tomu, že klouby robota jsou ve vertikální poloze, je robot lépe odolný vůči působení gravitace a tlaku vznikajícímu kvůli přenášení různých předmětů. [10]



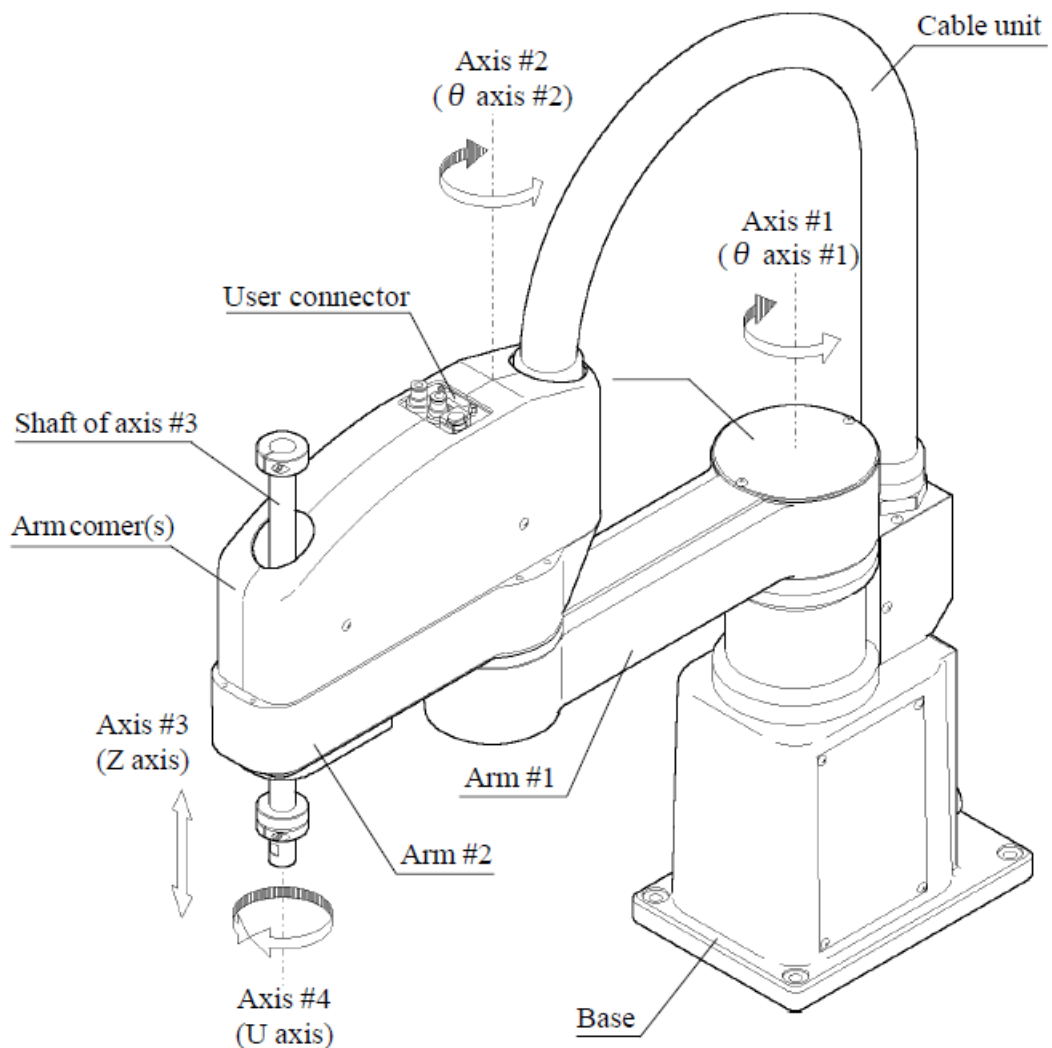
Obr. 1.6: Pracovní prostor SCARA robota [2]

## 1.5 Epson H554BN

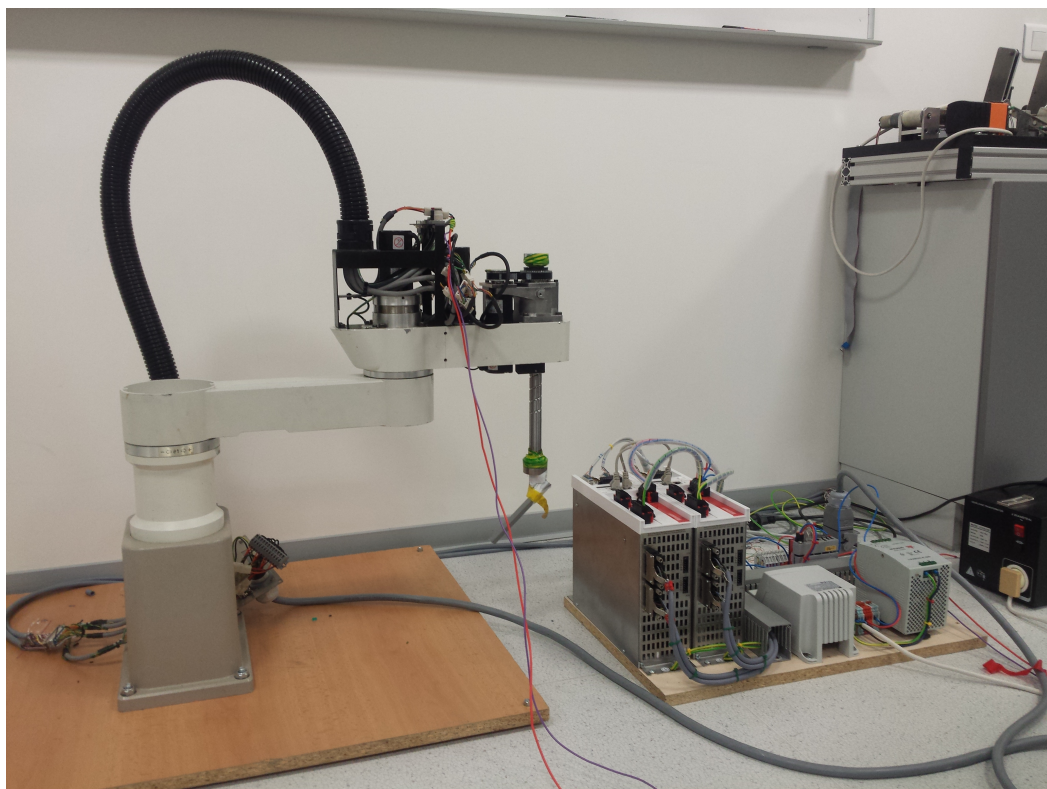
Jedná se o starší typ manipulátoru od firmy Epson, je to robot typu SCARA. První písmeno v označení manipulátoru určuje typ uchycení, další dvě číslice určují celkovou délku ramen, poslední číslice říká, kolik os manipulátor má. Poslední dvě písmena udávají jeho typ. Z tohoto označení lze zjistit, že manipulátor má horizontální uchycení, jeho celkový dosah všech ramen je 550 mm a má čtyři osy. První dvě osy jsou rotační a mají na starosti horizontální pohyb, třetí osa je translační a stará se o vertikální pohyb. Čtvrtá osa, která je rotační, dovoluje uchycený objekt otáčet podél osy Z. Kabeláž je vedena vnějškem a je vyvedena od zadní strany základny robota a přivedena na jeho druhé rameno. Vzhled manipulátoru s jeho popisem je zobrazen na obrázku 1.7. Celý manipulátor spolu s příslušenstvím je zobrazen na obrázku 1.8. Maximální hmotnost, se kterou může robot manipulovat, je do 5 kg, jedná se o součet hmotností přenášeného výrobku s hmotností pracovní



hlavice. Elektrické kabely jsou stíněné a maximální povolené napětí je 30 V stejnosměrných, nebo střídavých. Maximální povolený proud je 1 A. Manipulátor je také vybaven dvěma trubicemi o vnitřním rozměru 4 mm pro rozvod vzduchu. Tloušťka stěny vzduchové trubice je 2 mm a maximální povolený tlak vzduchu procházejícího trubicemi je 0,59 MPa. Všechny konce vzduchových trubic jsou vybaveny konektory pro připojení dalších vzduchových trubic o vnějším průměru 6 mm. Elektrické dráty a vzduchové trubice jsou vedené spolu v jedné kabelové chráničce. Rozsah pohybu manipulátoru je zobrazen na obrázku 1.9. Hvězdička ve schématu indikuje pozici mechanické zarážky. [11]



Obr. 1.7: Manipulátor Epson H554BN [11]



Obr. 1.8: Manipulátor Epson H554BN spolu s příslušenstvím

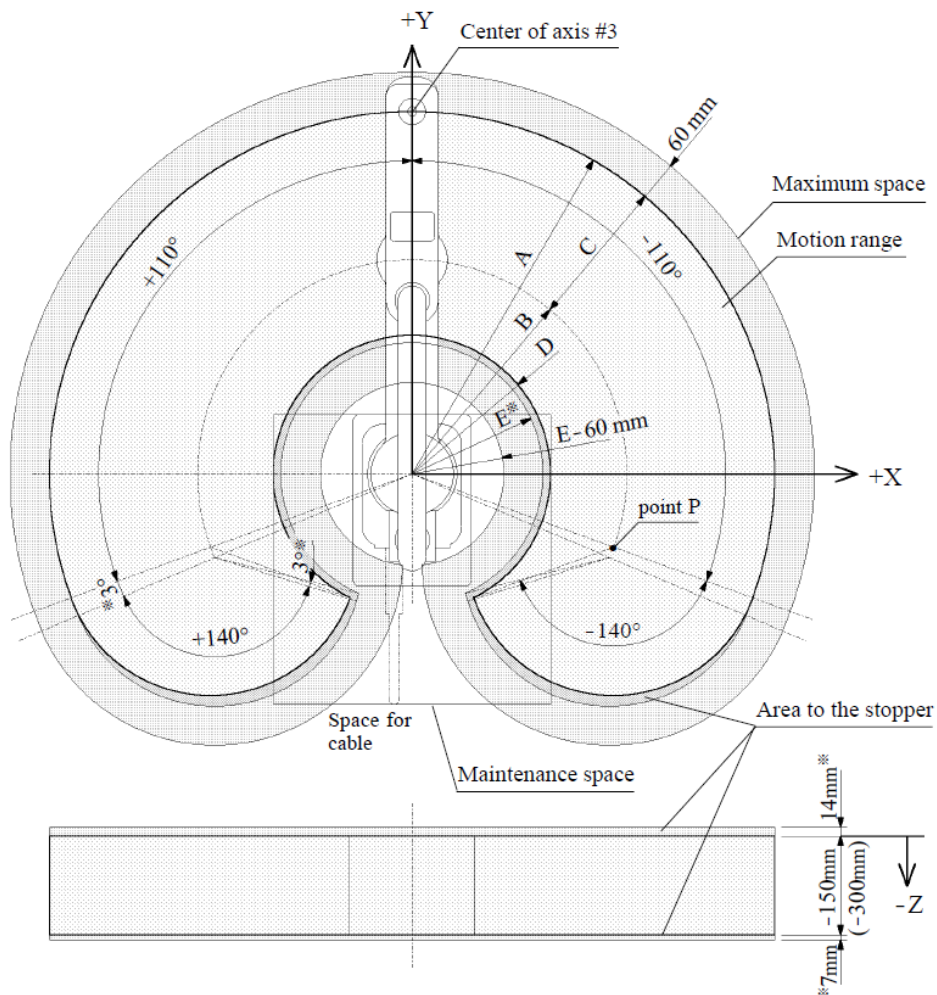
### 1.5.1 Pohon manipulátoru

Osy jsou poháněny synchronními servomotory od japonské firmy Yaskawa. Všechny servomotory jsou napájeny střídavým napětím o hodnotě 200 V. Pro transformaci síťového napětí na 200 V je použit autotransfornátor PVEA 2000 od firmy Tuffvassons. První osa je poháněna typem SGM-02A3SU12. Druhou osu pohání typ SGM-01A3SU11. Třetí osu pohání typ SGM-01A3SU21 a čtvrtá osa je poháněna typem SGM-A5A3SU11. V tabulce 1.1 jsou uvedeny parametry daných servomotorů, lze je určit podle prvních tří znaků za pomlčkou. Ke každému servomotoru je připojen inkrementální enkodér, který zjišťuje polohu dané osy. Pro osu 1 může enkodér nabývat hodnot pulzů od -36 409 do +364 089. Pro osu 2 může enkodér nabývat hodnot pulzů od -159 289 do +159 289. Pro osu 3 může enkodér nabývat hodnot pulzů od 0 do -92 160. Pro osu 4 může enkodér nabývat hodnot pulzů od -172 032 do +172 032. Převodový poměr pro první osu je 1:24 ( $4,369^\circ$  na jednu otáčku motoru). Převodový poměr pro druhou osu je 1:15 ( $7^\circ$  na jednu otáčku motoru). Převodový poměr pro třetí osu je 1:7,85 (13,333 mm na jednu otáčku motoru). Převodový poměr pro čtvrtou osu je 1:6 ( $17,476^\circ$  na jednu otáčku motoru). [11, 12]

Tab. 1.1: Parametry jednotlivých servomotorů [12]

Typ	02A	01A	A5A
Jmenovitý výkon [W]	200	100	50
Jmenovitý točivý moment [N*m]	0,637	0,318	0,159
Jmenovitý proud [A]	2,00	0,87	0,60
Jmenovité otáčky [ot/min]	3000	3000	3000

Standard motion range of the manipulator		H554BN
	A	550 mm
	B	325 mm
	C	225 mm
	D	211 mm
	E	198 mm



Obr. 1.9: Rozsah pohybu manipulátoru H554BN [11]

## **2 Řídicí systém a komponenty od firmy Beckhoff**

### **2.1 Servoměniče**

Napájení pro servomotory na ose 1 a 2 je vyvedeno ze servoměniče AX5203, zde jsou také přivedeny signály z enkodérů pro tyto dvě osy. Napájení pro servomotory na ose 3 a 4 je vyvedeno ze servoměniče AX5201, zde jsou také přivedeny signály z enkodérů pro tyto dvě osy. Jedná se o dvoukanálové servoměniče od firmy Beckhoff. První dvě písmena určují, o jaký typ zařízení se jedná. První číslice určuje jejich řadu, druhá číslice říká, kolik kanálů dané zařízení má. Poslední dvě číslice označují hodnotu jmenovitého proudu pro každý kanál. AX5201 má jmenovitý proud pro každý kanál 1,5 A a maximální jmenovitý proud pro trojfázové zapojení motorů může být 3 A. AX5203 má jmenovitý proud 3 A pro každý kanál, maximální jmenovitý proud může být až 6 A. Měniče obsahují svorkovnice pro připojení enkodérů a resolvérů, svorkovnice pro připojení senzorů pro měření teploty motorů, svorkovnice pro různé typy napájení, digitální vstupy a výstupy a vstup s výstupem pro EtherCAT. Tyto servoměniče používají vektorové řízení. [13]

#### **2.1.1 Vektorové řízení**

Oba servoměniče používají vektorové řízení, které rozděluje vazbu složek proudu, aby je mohlo řídit samostatně. Je řízen magnetický tok a moment motoru. Pro výpočet se použije náhradní obvod, tzv. matematický model. Díky vektorovému řízení se zlepšují dynamické vlastnosti motoru a přesnost. Lze rozlišit dva typy vektorového řízení - se zpětnou vazbou a bez zpětné vazby. Jelikož jsou k jednotlivým servomotorům připojeny enkodéry, tak se v tomto případě jedná o řízení se zpětnou vazbou. [14]

### **2.2 Komunikační rozhraní EtherCAT**

Beckhoff používá pro komunikaci mezi svými zařízeními jím vyvinutou technologii EtherCAT. Jedná se o velmi rychlou a výkonnou průmyslovou technologii, která spojuje různé řídicí systémy a zařízení. EtherCAT je mezinárodně uznávaný standard, který je volně k použití. Díky tomu se hodně rozšířil. Používá ho i hodně jiných firem pro sjednocení komunikace mezi svými zařízeními. Díky své rychlosti se využívá například pro měřicí zařízení, robotiku, elektrárny a další oblasti, které vyžadují přenos dat ve velmi rychlém čase. [15]

## 2.3 Coupler a terminály

Pro propojení jednotlivých terminálů se používá coupler EK1100. Tento coupler slouží k propojení technologie EtherCAT s libovolným množstvím různých terminálů. Pouze při připojení většího množství terminálů je nutné použít speciální napájecí terminály, jelikož maximální odběr proudu terminály je 2 A. EK1100 konvertuje procházející rámce z ethernetu 100BASE-TX na signály pro sběrnici E-bus, kterou jsou jednotlivé terminály spojeny. Pro manipulátor H554BN je použita stanice ve složení tohoto coupleru spolu se čtyřmi výstupními digitálními terminály EL2008 a dvěma vstupními digitálními terminály EL1124. EL2008 má osm digitálních výstupů, jejichž stav je indikován LED diodami přímo na daném terminálu. EL1124 má čtyři digitální vstupy, jejichž stav je indikován LED diodami přímo na daném terminálu. Z důvodu mechanické a elektrické ochrany je na konci terminálů připojen kryt EL9011. [16, 17, 18, 19]

## 2.4 Průmyslové počítače a operátorské panely

Beckhoff nabízí průmyslové počítače různých typů. Jeho embedded počítače mají označení CX a kombinují průmyslové počítače s modulárními zařízeními. Lze je namontovat na DIN lištu a připojit k nim různé terminály. Mezi další typ patří panelové počítače, které nesou označení CP. CP2xxx slouží pro připojení na přední panel rozvaděče a CP3xxx pro montáž na různé typy ramen. Obě série nabízí několik velikostí panelů a více dotykové (Multi-Touch) ovládání. Posledním typem jsou klasické průmyslové počítače přímo do rozvaděčů. V této oblasti Beckhoff nabízí různě velké a výkonné počítače, které nesou označení C5xxx nebo C6xxx. Průmyslové počítače od Beckhoffu mohou mít i přes identické označení jiný hardware, záleží na požadavcích zákazníka. Proto je nutné pro přesnou identifikaci daného počítače i jeho sériové číslo. Pro řízení manipulátoru H554BN je použit ultra-kompaktní průmyslový počítač C6015-0010. [20]

Počítač C6015-0010 má pasivní chlazení, jeho rozměry jsou 82 x 82 x 40 mm a je vhodný pro práci v teplotním rozsahu od 0 °C do +55 °C, díky tomu může být umístěn téměř kdekoliv. Počítač, který řídí manipulátor, je umístěn na panelu se servoměniči a ostatními zařízeními. Má šest konektorů - jeden pro napájení, dva síťové konektory RJ-45, jeden konektor DisplayPort, jeden port USB2.0 a jeden port USB3.0. Tento průmyslový počítač vyžaduje napájení 24 V stejnosměrných. Na jeho boku se nachází tři LED kontrolky. Jedna je pro pevný disk, jedna indikuje stav TwinCATu, což je řídicí software, a poslední kontrolka zobrazuje stav počítače. Počítač má procesor Intel Atom E3827, pevný disk SSD 30 GB a 4 GB paměť RAM. Operační systém uvnitř počítače je Windows 10 IoT Enterprise 2016. [21]

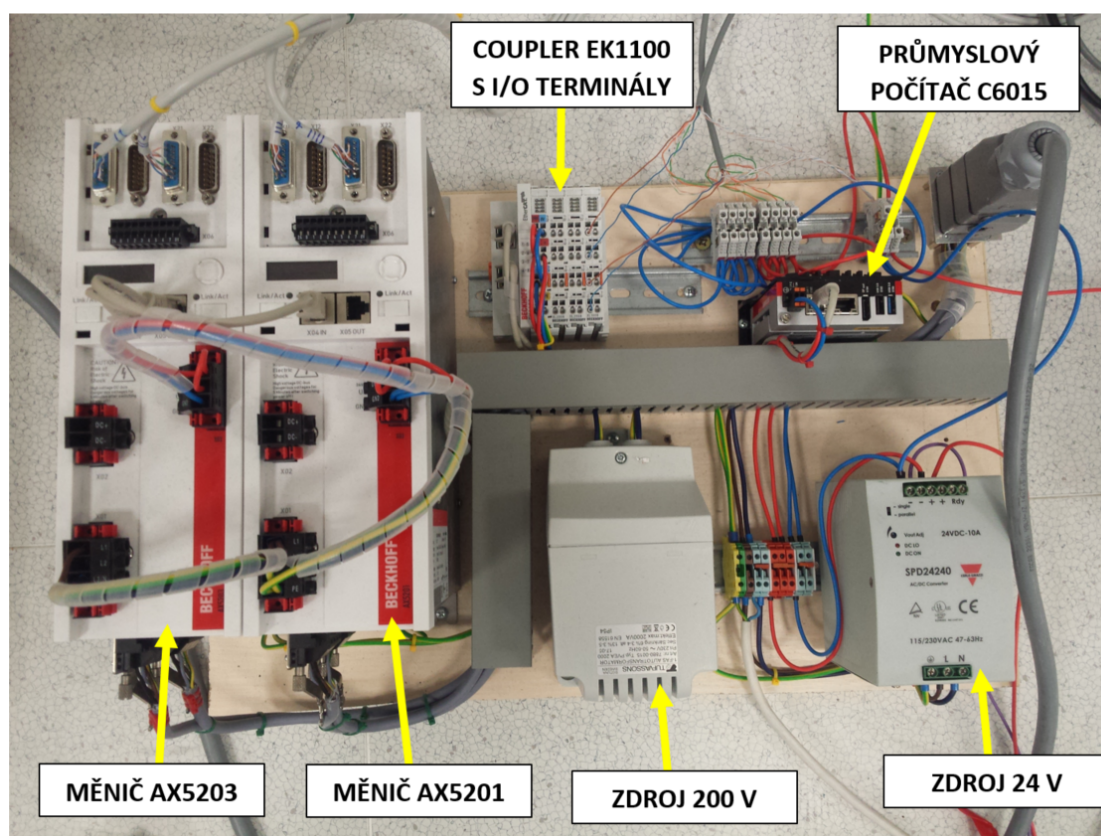


Operátorský panel má označení CP29xx pro umístění na přední panel rozvaděče a CP39xx pro umístění na různé typy ramen. Při připojení panelu přes port USB nebo DVI může být panel vzdálen od řídicího počítače do 50 metrů. Při připojení přes speciální technologii CP-Link 4 může být vzdálenost navýšena na 100 metrů. [20]

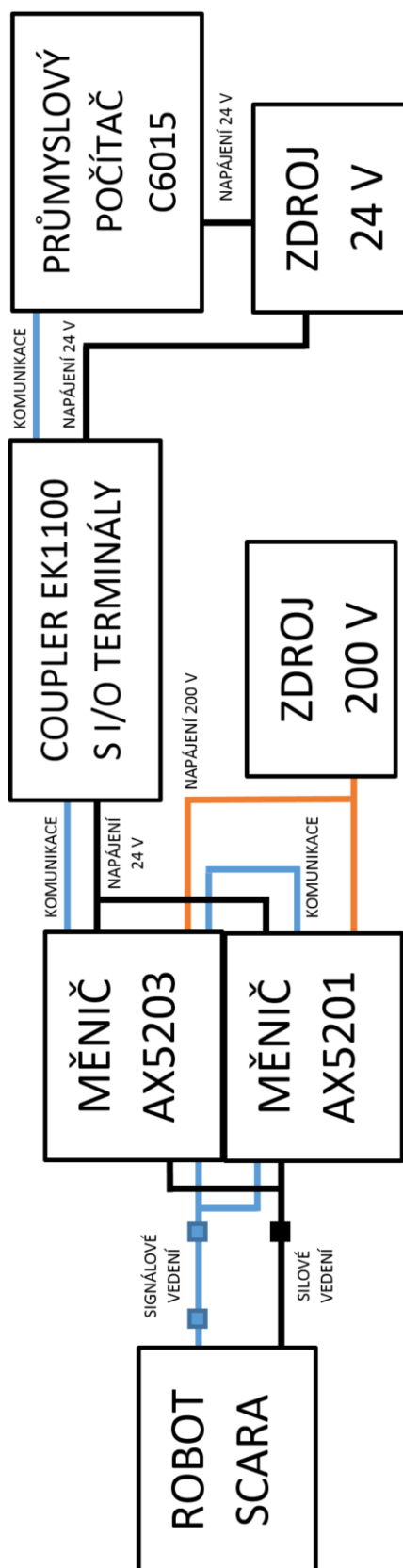
Řídicí systém musí mít svou licenci. Licencuje se vždy až po kompletním zprovoznění celého systému. Licenci lze uložit přímo v průmyslovém počítači, ale při zničení počítače dojde k její ztrátě, proto se pro licencování doporučuje použít speciální terminál EL6070. Lze také použít speciální flash disk. [22]

Pro napájení zařízení, které požadují stejnosměrné napětí je použit AC/DC měnič SPD24240 od firmy Carlo Gavazzi.

Na obrázku 2.1 je zobrazen panel, na kterém jsou umístěna jednotlivá zařízení. Na obrázku 2.2 je zobrazené blokové schéma zapojení řídicího systému se SCARA robotem.



Obr. 2.1: Panel se zařízeními pro ovládání manipulátoru



Obr. 2.2: Blokové schéma zapojení řídicího systému se SCARA robotem

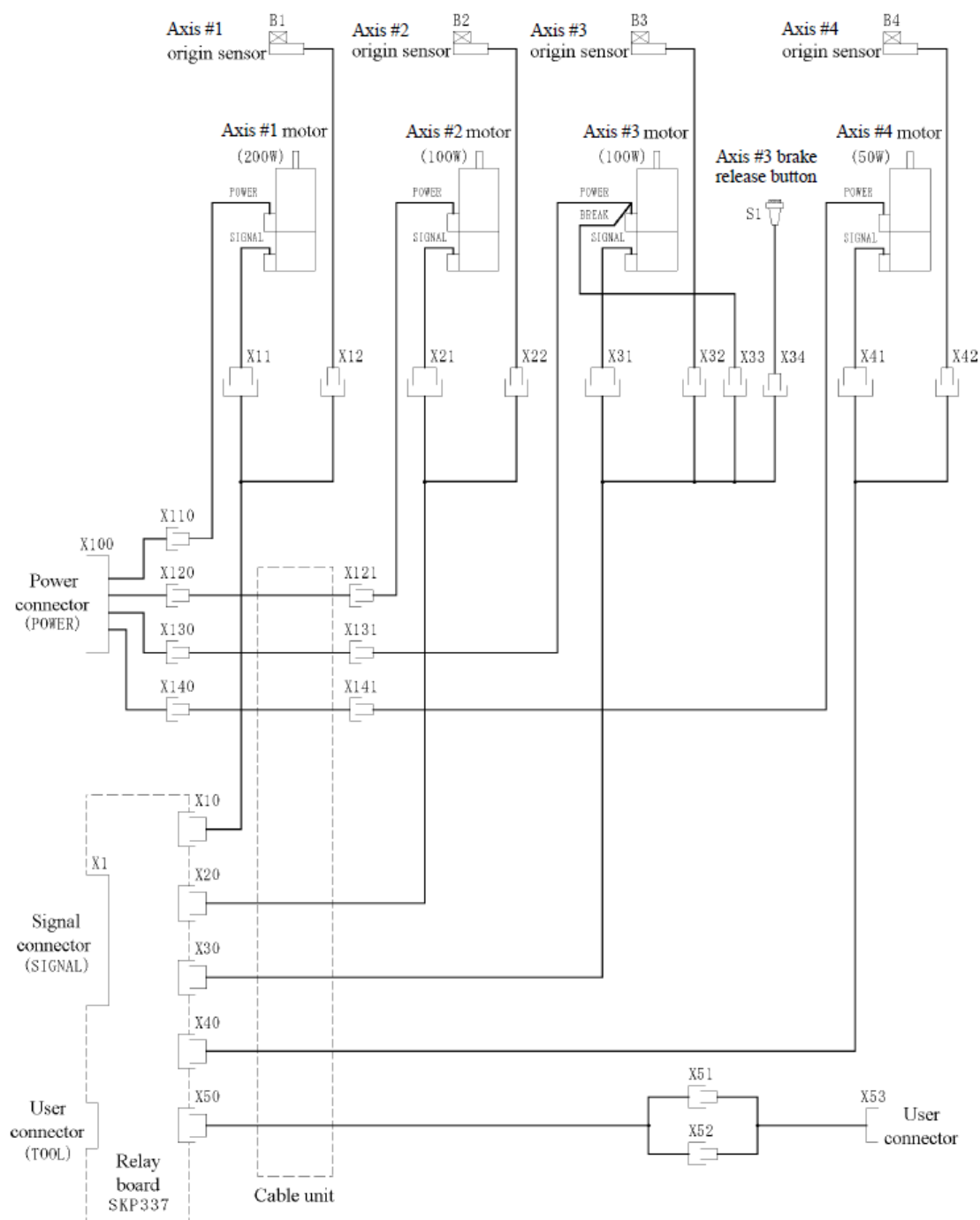
### 3 Elektronické zapojení

Téměř celý manipulátor včetně jeho příslušenství byl již zapojen. Napájení jednotlivých servomotorů je vyvedeno přes svorkovnice X110, X120, X130 a X140 na svorkovnici X100, což je napájecí konektor na zadní straně základny manipulátoru. První číslice v označení všech svorkovnic říká, o kterou osu se jedná. Podle původního zapojení byly signály a napájení z jednotlivých enkodérů vyvedeny přes svorkovnice X11, X21, X31 a X41. Každá osa má navíc svůj senzor pro koncovou polohu, kterého signálové a napájecí dráty vedou přes svorkovnice X12, X22, X32 a X42. Osa 3 má navíc svorkovnice X33 pro brzdu a X34 pro tlačítko sloužící k uvolnění brzdy. Všechny signály a napájení z uvedených svorkovnic, kromě svorkovnic pro napájení servomotorů, byly přivedeny na svorkovnice X10, X20, X30 a X40. Tyto čtyři svorkovnice byly připájeny na desku robota nacházející se na zadní straně základny. Na tuto desku byla připájena i svorkovnice X50, ze které vedly signály přes svorkovnice X51 a X52 na svorkovnici X53, což je uživatelský konektor nacházející se na druhém rameni manipulátoru. Z desky byl vyveden signálový konektor, který se propojil 68 žilovým kabelem s deskou umístěnou na panelu s měniči a dalším příslušenstvím. Na této desce bylo nutné zjistit, který signál je kde přiveden, a propojit signály z jednotlivých enkodérů se servoměniči AX5201 a AX5203, které slouží k ovládání jednotlivých servomotorů. Zjednodušené schéma zapojení jednotlivých svazků je zobrazeno na obrázku 3.1. [23]

Při zjišťování, který signál je kde přiveden, jsem objevil zkrat na desce umístěné v základně manipulátoru a zároveň jsem nemohl najít většinu signálů z enkodérů. Při hledání chyby a rozebrání propojujícího kabelu jsem zjistil, že je poškozena velká část drátů. Proto se po konzultaci s mým vedoucím rozhodlo, že se deska umístěná v manipulátoru, včetně připájených svorkovnic X10, X20, X30, X40 a X50 odstraní a místo původního propojovacího kabelu se použije několik UTP kabelů CAT 5E, které se na jedné straně připájí přímo na odstřižené dráty z výše uvedených svorkovnic, a na druhé straně se připojí konektory RJ-45. Ze servoměničů byly již vyvedeny UTP kabely CAT 5E, na které bylo potřeba připojit samice konektorů typu RJ-45.

Nejdříve jsem vytvořil dvoumetrové kabely pro připájení k jednotlivým drátům od strany robota a napojil na ně konektory, poté jsem postupně začal pájet jednotlivé dráty k robotovi. Po absolvování školení ve firmě Beckhoff v prosinci 2019 jsem zjistil, že použitá kabeláž není vhodná, jelikož není stíněná, a že je potřeba použít stínění pro všechny kabely vedoucí do servoměničů. Při použití nestíněných kabelů je velká pravděpodobnost, že dojde k rušení. V důsledku toho například u servomotorů nebude možné přes připojené enkodéry určit jejich přesnou polohu. Rozhodlo se proto, že v uvedené opravě budu pokračovat, ale dodatečně se všechny kabely

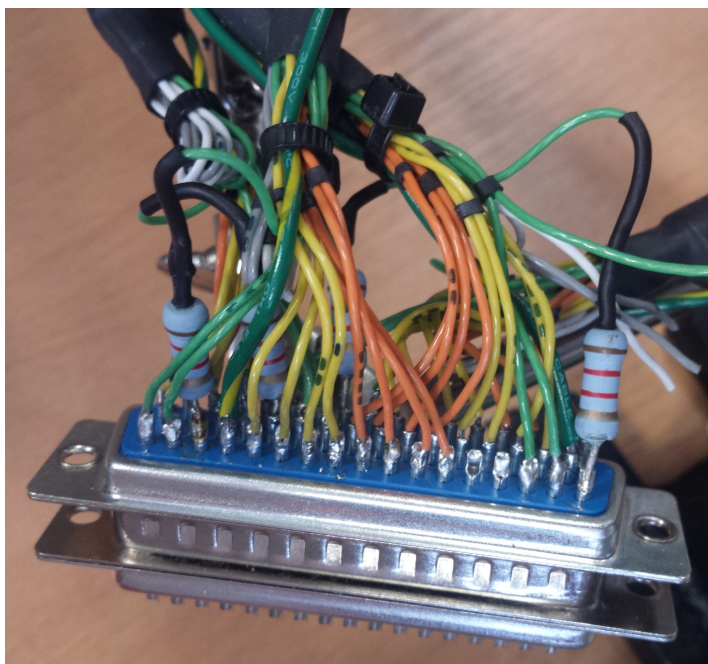




Obr. 3.1: Zjednodušené schéma zapojení H554BN [23]

přistíní. Původně bylo v plánu pro samice konektorů RJ-45 vytvořit novou desku a konektory tam připájet spolu s dráty vyvedenými ze servoměničů, ale zjistilo se, že tyto konektory nemívají přesně umístěné jednotlivé piny. Vytvořit novou desku by kvůli tomu bylo obtížné, proto se rozhodlo použít jiný způsob opravy. Pořídily se

dva dvoumetrové 25 žilové kabely a nové konektory D-SUB CAN 50. Připájené kabely CAT 5E jsem odstříhl a připájel jednotlivé dráty k novým konektorům. Použil jsem čtyři konektory, aby bylo možné propojovací kabely jednoduše odpojit jak od robota, tak i od servoměničů. Ke každému signálu HOME z jednotlivých senzorů pro koncovou polohu bylo nutné navíc připájet rezistory o hodnotě  $1,2\text{ k}\Omega$ . Tyto rezistory jsem odpájel z vyzkratované desky a připájel jsem je k prvnímu konektoru od strany robota. Vzhled tohoto konektoru se nachází na obrázku 3.2. Během pájení jsem zjistil, že u enkodérů pro osy 2-4 nefunguje signál FGND, což znamená, že je přerušené stínění kabelů uvnitř manipulátoru. [23]



Obr. 3.2: Konektor D-SUB CAN 50

Podrobné schéma zapojení, kromě propojovacího kabelu mezi deskou v manipulátoru a servoměniči, se nachází v příloze A, jedná se o schéma zapojení před provedenou opravou. Upravená část schématu zapojení podle provedené opravy se nachází v příloze B, v tomto schématu se nachází již pouze změny oproti původnímu zapojení. Schéma zapojení jsem nakreslil v softwaru WSCAD SUITE Education.

### 3.1 Software WSCAD

Tento software obsahuje části Electrical Engineering, Cabinet Engineering, Piping & Instrumentation, Fluid Engineering, Building Automation a Electrical Installation. Slouží pro tvorbu různých druhů schémat. Kromě tvorby elektronických schémat ho

lze použít například pro návrh rozložení různých zařízení v továrně, návrh elektrické instalace v budovách, k plánování a tvorbě dokumentace pro pneumatické a hydraulické zařízení atd. Pro tvorbu elektronického schématu jsem použil část Electrical Engineering. Jednotlivá zařízení nakreslená ve schématech jsem si z větší části vytvořil sám, několik zařízení již existovalo a stačilo je přidat z knihoven. [24]

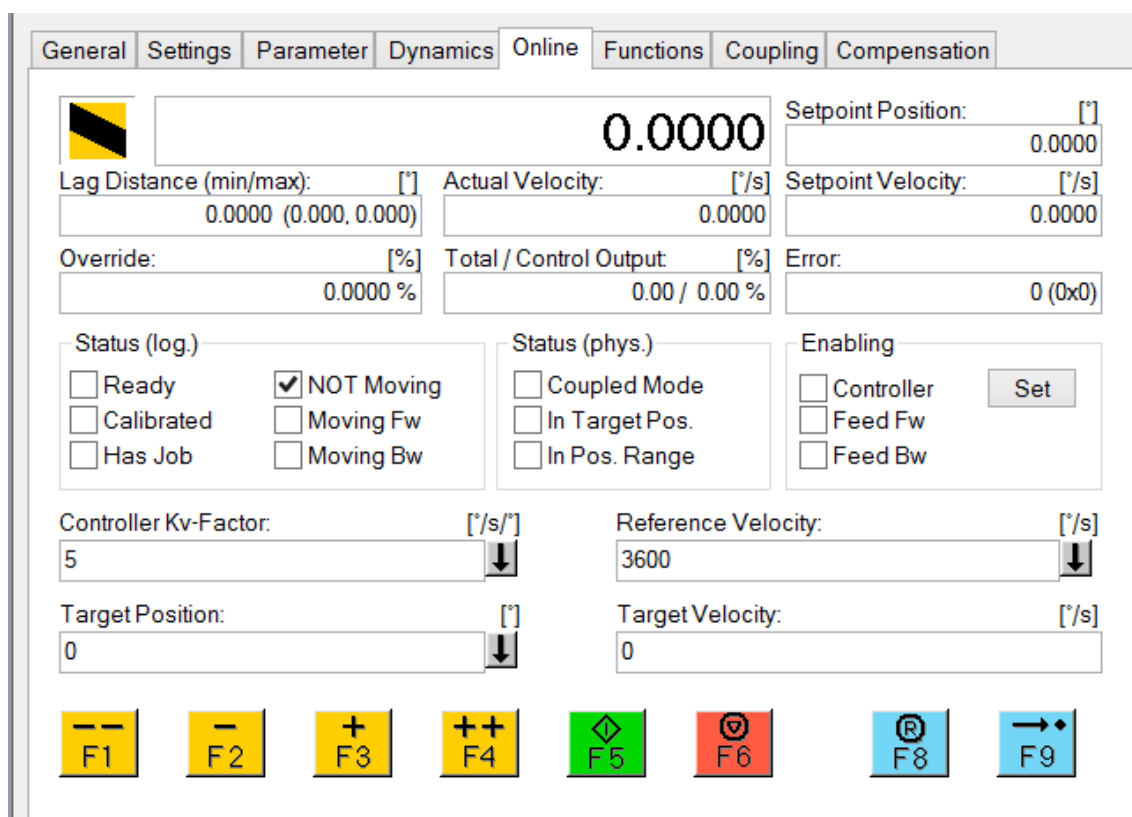
## 4 Řízení manipulátoru Epson H554BN

Manipulátor se programuje v řídicím softwaru TwinCAT 3 od firmy Beckhoff. Pro práci s tímto softwarem je nutné mít nainstalované vývojové prostředí Visual Studio 2017 od společnosti Microsoft. V době, kdy jsem vytvářel základní funkce pro pohyb a řízení jednotlivých os samostatně, manipulátor H554BN ještě nebyl kompletně opraven, proto ho nebylo možné naprogramovat a otestovat na něm tyto funkce. Z tohoto důvodu jsem níže napsaný postup konfigurace včetně funkcí pro pohyb nejdříve otestoval na hardwaru ve firmě Beckhoff v rámci jejich školení a poté jsem po opravení manipulátoru provedl identický postup i na něm. Při programování manipulátoru jsem již nepoužil funkční bloky pro pohyb jednotlivých motorů samostatně, ale řešil jsem pohyb již pomocí kinematické transformace.

### 4.1 Počáteční konfigurace

Před samotným programováním pohybů robota je nutné nejdříve načíst všechna připojená zařízení, což jsem provedl pomocí funkce Scan, která je v sekci I/O - Devices. Když se po naskenování zařízení objevil dotaz, kde se mají připojit načtené osy, zvolil jsem NC konfiguraci. NC je zkratka pro numerical control (numerické řízení).[25] Dále je nutné správně nastavit parametry motorů pro jednotlivé osy přes Drive Manager, ten se stáhne ze stránek Beckhoffu. Přes Drive Manager se také nastavují různé typy regulátorů pro motory. Všechny nastavené parametry jsem poté uložil do záložky Startup List, která slouží jako výchozí nastavení. U enkodéru jsem vypočítal a nastavil, o kolik milimetrů na lineární ose se motor posune za jeden pulz. Vydělí se vzdálenost, kterou motor zvládne urazit za jednu otáčku rozlišením enkodéru v TwinCATu.

S tímto nastavením lze již otestovat funkčnost motorů. Po aktivaci konfigurace a spuštění se povolí panel, který je zobrazen na obrázku 4.1. Je nutné ještě povolit ovládání tlačítkem Set. Pro otáčení hřídele motoru slouží žlutá tlačítka, lze si vybrat ze dvou rychlostí otáčení. Pomalejší otáčení je značeno [+] a [-], rychlejší otáčení je [++] a [--]. Zelené tlačítko slouží k povolení otáčení a červené k jeho zastavení. Lze otestovat i pohyb na určitou pozici, to se udělá v záložce Functions. Modré tlačítko s písmenem [R] slouží pro reset případné chyby. Modré tlačítko F9 je pro přepínání breakpointů.



Obr. 4.1: Panel pro testování funkčnosti motorů

## 4.2 Komutační offset

Komutace popisuje změnu směru proudu ve vinutí, které ovlivňuje důležité charakteristiky servomotorů, například jeho účinnost, plynulý chod a optimální tepelné využití. Jelikož manipulátor nepoužívá motory od firmy Beckhoff, bylo potřeba manuálně nastavit komutační offset. [26]

## 4.3 Definování a přiřazení proměnných

Pro psaní kódu jsem přidal do existujícího projektu novou část TwinCAT PLC. V sekci GVLs jsem vytvořil nový objekt Global Variable List, do kterého jsem nadefinoval všechny globální proměnné. První písmena v názvu proměnných určují, o jaký typ proměnné se jedná. Přiřazení do proměnných jsem provedl v sekci MAIN, která se volá po spuštění programu a je v záložce POU (Program Organization Unit - organizační jednotka programu do které patří funkce, funkční bloky a objekty typu program), zde lze také nadefinovat lokální proměnné. [27] Do záložky POU jsem následně vložil funkční bloky sloužící k ovládání motorů, za účelem testování

základních funkcí pro pohyb na hardwaru ve firmě Beckhoff. Při tvorbě kódu pro manipulátor jsem záložku POUs použil pro objekty obsahující jednotlivé části potřebné pro kinematickou transformaci. Pro přiřazení proměnných na vstupy a výstupy je nutné spojit je s jednotlivými vstupy a výstupy. To jsem udělal pomocí položky Change Link u vstupů a výstupů jednotlivých terminálů. Přiřazení os lze změnit v jejich nastavení pomocí položky Link To I/O.

## 4.4 Funkční bloky pro ovládání motorů

Abych mohl používat funkční bloky pro řízení motorů, musel jsem přidat do sekce References knihovnu Tc2\_MC2. Podle jejího názvu se jedná o knihovnu vytvořenou pro TwinCAT 2, ale je funkční i v prostředí TwinCAT 3.

### 4.4.1 MC\_Power

Aktivuje softwarové povolení pro osu. Funkční blok je zobrazen na výpisu 4.1. [28]

Výpis 4.1: Funkční blok MC\_Power

```
1 fbPower(  
2   Axis:= GVL.Axis1,  
3   Enable:= bEnable,  
4   Enable_Positive:= TRUE,  
5   Enable_Negative:= TRUE,  
6   Override:= 100.0,  
7   BufferMode:= MC_Aborting,  
8   Options:= ,  
9   Status=> ,  
10  Busy=> ,  
11  Active=> ,  
12  Error=> ,  
13  ErrorID=> );
```

Popis parametrů funkčního bloku: [28]

- Axis - výběr osy
- Enable - obecné softwarové povolení osy
- Enable\_Positive - povolení posuvu vpřed, funguje pouze, pokud je Enable ve stavu true
- Enable\_Negative - povolení posuvu vzad, funguje pouze, pokud je Enable ve stavu true
- Override - změna rychlosti v procentech pro všechny ovládací funkční bloky

- BufferMode - vyhodnocuje se při resetu povelu Enable; MC\_Aborting ihned deaktivuje osu
- Options - dodatečné parametry
- Status - nastaví se na true, když je osa připravena
- Busy - je nastaveno na true po celou dobu kdy Enable = true
- Active - označuje provedení příkazu
- Error - nastaví se na true, pokud nastane nějaká chyba
- ErrorID - pokud vznikne chyba, tak tento parametr vypíše její identifikační kód

#### 4.4.2 MC\_Reset

Resetuje NC osu. Funkční blok je zobrazen na výpisu 4.2. [29]

Výpis 4.2: Funkční blok MC\_Reset

```

1 fbReset(
2   Axis:= GVL.Axis1,
3   Execute:= bExecute,
4   Done=> ,
5   Busy=> ,
6   Error=> ,
7   ErrorID=> );

```

Popis parametrů funkčního bloku: [29]

- Axis - výběr osy
- Execute - spuštění na náběžnou hranu
- Done - nastaví se na true po úspěšném resetu
- Busy - je nastaveno na true po celou dobu běhu funkčního bloku
- Error - nastaví se na true, pokud nastane nějaká chyba
- ErrorID - pokud vznikne chyba, tak tento parametr vypíše její identifikační kód

#### 4.4.3 MC\_Stop

Zastaví osu s nadefinovanou brzdou rampou a zablokuje ji proti spuštění jiných funkčních bloků pro pohyb. Funkční blok je zobrazen na výpisu 4.3. [30]

Výpis 4.3: Funkční blok MC\_Stop

```

1 fbStop(
2   Axis:= GVL.Axis1,
3   Execute:= bExecute,

```

```

4   Deceleration:= 0.0,
5   Jerk:= 0.0,
6   Options:= ,
7   Done=> ,
8   Busy=> ,
9   Active=> ,
10  CommandAborted=> ,
11  Error=> ,
12  ErrorID=> );

```

Popis parametrů funkčního bloku: [30]

- Axis - výběr osy
- Execute - spuštění na náběžnou hranu
- Deceleration - brzdění; pokud je nastavená hodnota menší nebo rovno 0, tak se použije brzdění nastavené v poslední použité funkci Move
- Jerk - brzdění v čase; pokud je nastavená hodnota menší nebo rovno 0, tak se použije brzdění v čase nastavené v poslední použité funkci Move
- Options - dodatečné parametry
- Done - nastaví se na true po kompletním zastavení
- Busy - je nastaveno na true po celou dobu běhu funkčního bloku
- Active - označuje, že funkční blok ovládá osu; zůstává ve stavu true tak dlouho, dokud není Execute false
- CommandAborted - nastaví se na true, pokud funkční blok nemohl být úplně dokončen
- Error - nastaví se na true, pokud nastane nějaká chyba
- ErrorID - pokud vznikne chyba, tak tento parametr vypíše její identifikační kód

#### 4.4.4 MC\_MoveAbsolute

Spustí pohyb na nadefinovanou absolutní pozici. Funkční blok je zobrazen na výpisu 4.4. [31]

Výpis 4.4: Funkční blok MC\_MoveAbsolute

```

1  fbAbs(
2    Axis:= GVL.Axis1,
3    Execute:= bExecute,
4    Position:= fPosition,
5    Velocity:= fVelocity,
6    Acceleration:= 0.0,

```



```

7   Deceleration:= 0.0,
8   Jerk:= 0.0,
9   BufferMode:= MC_Aborting,
10  Options:= ,
11  Done=> ,
12  Busy=> ,
13  Active=> ,
14  CommandAborted=> ,
15  Error=> ,
16  ErrorID=> );

```

Popis parametrů funkčního bloku: [31]

- Axis - výběr osy
- Execute - spuštění na náběžnou hranu
- Position - cílová pozice
- Velocity - maximální rychlost
- Acceleration - zrychlení; pokud je nastavená 0, tak se použije zrychlení nastavené v konfiguraci osy
- Deceleration - brzdění; pokud je nastavená 0, tak se použije brzdění nastavené v konfiguraci osy
- Jerk - zrychlení v čase; pokud je nastavená 0, tak se použije zrychlení v čase nastavené v konfiguraci osy
- BufferMode - vyhodnocuje se, pokud osa již vykonává jiný příkaz
- Options - dodatečné parametry
- Done - nastaví se na true po dosažení cílové pozice
- Busy - je nastaveno na true po celou dobu běhu funkčního bloku
- Active - označuje provedení příkazu; pokud byl příkaz ve vyrovnávací paměti, stane se aktivní až po dokončení spuštěného příkazu
- CommandAborted - nastaví se na true, pokud funkční blok nemohl být úplně dokončen
- Error - nastaví se na true, pokud nastane nějaká chyba
- ErrorID - pokud vznikne chyba, tak tento parametr vypíše její identifikační kód

#### 4.4.5 MC\_MoveRelative

Spustí posuv o určitou vzdálenost od aktuální vzdálenosti. Funkční blok je zobrazen na výpisu 4.5. [32]

#### Výpis 4.5: Funkční blok MC\_MoveRelative

```
1 fbRel(  
2   Axis:= GVL.Axis1,  
3   Execute:= bExecute,  
4   Distance:= fDistance,  
5   Velocity:= fVelocity,  
6   Acceleration:= 0.0,  
7   Deceleration:= 0.0,  
8   Jerk:= 0.0,  
9   BufferMode:= MC_Aborting,  
10  Options:= ,  
11  Done=> ,  
12  Busy=> ,  
13  Active=> ,  
14  CommandAborted=> ,  
15  Error=> ,  
16  ErrorID=> );
```

Popis parametrů funkčního bloku: [32]

- Axis - výběr osy
- Execute - spuštění na náběžnou hranu
- Distance - relativní vzdálenost, o kterou dojde k posuvu
- Velocity - maximální rychlost
- Acceleration - zrychlení; pokud je nastavená 0, tak se použije zrychlení nastavené v konfiguraci osy
- Deceleration - brzdění; pokud je nastavená 0, tak se použije brzdění nastavené v konfiguraci osy
- Jerk - zrychlení v čase; pokud je nastavená 0, tak se použije zrychlení v čase nastavené v konfiguraci osy
- BufferMode - vyhodnocuje se, pokud osa již vykonává jiný příkaz
- Options - dodatečné parametry
- Done - nastaví se na true po dosažení cílové pozice
- Busy - je nastaveno na true po celou dobu běhu funkčního bloku
- Active - označuje provedení příkazu; pokud byl příkaz ve vyrovnávací paměti, stane se aktivní až po dokončení spuštěného příkazu
- CommandAborted - nastaví se na true, pokud funkční blok nemohl být úplně dokončen
- Error - nastaví se na true, pokud nastane nějaká chyba
- ErrorID - pokud vznikne chyba, tento parametr vypíše její identifikační kód

## 4.5 Kinematické řízení - počáteční konfigurace

Pro implementaci kinematického řízení robota bylo potřeba stáhnout ze stránek Beckhoffu speciální balíček pro kinematickou transformaci a nainstalovat ho na řídicí počítač C6015-0010 a počítač, který sloužil pro tvorbu programu pro manipulátor. Beckhoff rozděluje kinematickou transformaci na čtyři úrovně v závislosti na počtu os. Pro SCARA robota je potřebný balíček třetí úrovně TF5112, který je součástí balíčku TF5400-Advanced-Motion-Pack. Po instalaci balíčku jsem přidal do projektu knihovnu Tc2\_NcKinematicTransformation, která obsahuje potřebné prvky pro výpočet kinematiky. [33]

### 4.5.1 Nastavení os

Fyzické (hardwarové) osy načtené pomocí funkce Scan jsem vložil do podskupiny ACS. ACS je zkratka pro axis coordinate system (souřadnicový systém osy) a popisuje polohu fyzických os robota. [33] První, druhou a čtvrtou osu jsem nastavil ve stupních a třetí osu v milimetrech.

Kvůli kinematickému řízení jsem vytvořil čtyři nové osy - X, Y, Z a RZ, které jsem vložil do podskupiny MCS. Zkratka MCS znamená machine coordinate system (souřadnicový systém stroje). Jedná se o kartézský souřadnicový systém, který, v mém případě, má svůj počátek v první ose. [33] Osa RZ je určena pro čtvrtou osu. Tyto osy jsou čistě softwarové.

Všechny osy jsem propojil s částí PLC, aby bylo možné pomocí kinematické transformace převést systém ACS na systém MCS a zpět. Mimo tyto systémy je ještě možné použít WCS a UCS. WCS je world coordinate system (světový souřadnicový systém) a UCS je user coordinate system (uživatelský souřadnicový systém). [33]

### 4.5.2 Nastavení parametrů pro kinematické řízení

Po vytvoření daného typu kinematické skupiny bylo potřeba do ní zadat parametry robota. Seznam parametrů je zobrazen na obrázku 4.2.

Pro SCARA robota jsou nejdůležitější tyto parametry:

- Inner arm length;
- Outer arm length;
- Gear coupling;
- Flange Angle A;
- Tool offset OID.

**Inner arm length** je délka prvního ramene. Uvádí se od středu první osy po střed druhé osy. Manipulátor H554BN má délku prvního ramene 325 mm. [33]

Object	Context	Parameter (Init)	Interfaces				
	Name	Value	CS	Unit	Type	PTCID	
-	Configuration						
-	MCS offset	...	<input type="checkbox"/>			0x05010069	
	x-shift	0.0		mm	LREAL		
	y-shift	0.0		mm	LREAL		
	z-shift	0.0		mm	LREAL		
-	Spatial reference definition	...	<input type="checkbox"/>			0x05010100	
	.Translation X	0.0		mm	LREAL		
	.Translation Y	0.0		mm	LREAL		
	.Translation Z	0.0		mm	LREAL		
	.Rotation 1	0.0		°	LREAL		
	.Rotation 2	0.0		°	LREAL		
	.Rotation 3	0.0		°	LREAL		
	.Rotation convention	Rotation_Z3Y2X1_DIN9300			MC.CoordInterpretation_SO3		
	.Spatial reference	00000000			OTCID		
	.Definition direction	toReference			MC.ReferenceDefDir		
-	TCP extension						
	Flange angle A	220.0	<input type="checkbox"/>	°	LREAL	0x0501006D	
	Tool offset OID	00000000	<input type="checkbox"/>		OTCID	0x0501007C	
-	ACStoDOF coupling						
	Gear coupling	0.05555555556	<input type="checkbox"/>	mm / °	LREAL	0x05010080	
-	Kinematic						
	Inner arm length	325.0	<input type="checkbox"/>	mm	LREAL	0x05010020	
	Outer arm length	225.0	<input type="checkbox"/>	mm	LREAL	0x05010021	

Obr. 4.2: Parametry kinematické skupiny

**Outer arm length** je délka druhého ramene. Uvádí se od středu druhé osy po střed čtvrté osy. Délka druhého ramene manipulátoru H554BN je 225 mm. [33]

**Gear coupling** určuje poměr mezi třetí a čtvrtou osou. Při pohybu jedné z těchto os se mění zároveň poloha druhé z těchto os. Při otočení čtvrté osy o 360° se posune třetí osa o 2 cm, výsledný poměr je poté 0,055 56 mm/°. [33]

**Flange Angle A** stanoví úhel natočení kolem osy X (první osy). První osa se může otáčet o 220°. [33]

**Tool offset OID** slouží pro zadání identifikačního čísla použité pracovní hlavy. Manipulátor H554BN aktuálně nemá namontovanou žádnou pracovní hlavici, proto jsou v tomto parametru samé nuly. [33]

Další nastavení je již obecné pro jakýkoliv způsob kinematického řízení:

- **MCS offset** - Pomocí parametrů u této položky lze posunout počátek první osy vzhledem k souřadnicovému systému stroje (MCS). Například u SCARA robota lze pomocí parametru z-shift přesunout počátek ze středu první osy na základnu robota. [33]
- **Spatial reference definition** - Nastavením parametrů u této položky lze přesu-

nout a natočit souřadnicový systém stroje (MCS) v referenčním souřadnicovém systému. Pomocí parametrů Translation X-Z lze přesunout systém v jednotlivých směrech, hodnota se udává v milimetrech. Pomocí parametrů Rotation 1-3 se natáčí ve stupních jednotlivé úhly, pořadí úhlů je definované pomocí parametru Rotation convention. Parametr Spatial reference udává, který souřadnicový systém je použit jako základ pro souřadnicový systém stroje (MCS), hodnota 0 udává, že se jedná o světový souřadnicový systém (WCS). Definition direction určuje směr posunutí (z referenčního systému, nebo ze systému MCS). [33]

Po nastavení všech parametrů lze kinematickou transformaci aktivovat pomocí PLC. Pro spuštění transformace bylo ještě potřeba definovat cyklické kanálové rozhraní v PLC a propojit ho se vstupy a výstupy kinematického kanálu, to jsem provedl pomocí příkazů uvedených na výpisu 4.6.

Výpis 4.6: Propojení cyklického kanálového rozhraní s I/O kinematického kanálu

1	<code>in_stKinToPlc</code>	<code>AT %I*</code>	<code>: NCTOPLC_NCICHANNEL_REF ;</code>
2	<code>out_stPlcToKin</code>	<code>AT %Q*</code>	<code>: PLCTONC_NCICHANNEL_REF ;</code>

## 4.6 Implementace kinematického řízení

Zdrojový kód programu pro kinematické řízení jsem rozdělil do pěti objektů:

- Main;
- Axes\_control\_ACS;
- Axes\_control\_MCS;
- ACS\_Axes\_ActualPosition;
- Kinematika\_transformace.

Všechny objekty jsou typu program. Typ program dokáže během zpracování vrátit více hodnot. Všechny hodnoty se uchovávají až do příštího spuštění. Objekt typu program lze volat z různých objektů kromě funkcí. Program je oproti funkčnímu bloku rozdílný v tom, že pokud nějaký objekt zavolá program a tím změní parametry uvnitř programu, tak se tyto parametry zachovávají, i přes jeho opakované volání z jiného objektu. Ve funkčním bloku se zachovávají parametry pouze při volání identické instance. [27]

### 4.6.1 Main

Main byl vytvořen automaticky a spouští se v pravidelných intervalech. Přiřazují se zde hodnoty jednotlivým proměnným, které slouží k ovládaní všech os, jako je například softwarové povolení os, resetování os a jejich zastavení. Všechny proměnné

jsou globální a jsou uloženy v globálním seznamu proměnných GVL1. Main využívám také k ověření připravenosti ACS a MCS os k dalším operacím a k volání ostatních objektů.

#### **4.6.2 Axes\_control\_ACS**

Axes\_control\_ACS obsahuje funkční bloky MC\_Power, MC\_Reset a MC\_Stop sloužící k ovládaní fyzických os robota. Jednotlivé funkční bloky jsou detailněji popsány v kapitole 4.4. Osy lze řídit samostatně a nebo všechny zároveň. Objekt si zároveň ověřuje, zda jsou všechny ACS osy softwarově povolené a ukládá výsledný stav do globální proměnné bAcsAxesReady, která je poté vyčtena v objektu Main. Všechny potřebné proměnné pro funkční bloky získává objekt ze seznamu globálních proměnných GVL1.

#### **4.6.3 Axes\_control\_MCS**

Axes\_control\_MCS slouží k softwarovému povolení MCS os v kartézském souřadnicovém systému. Objekt používá funkční bloky MC\_Power, MC\_Reset a MC\_Stop, tyto funkční bloky jsou popsány v kapitole 4.4. Objekt zároveň kontroluje stav jednotlivých os a ukládá ho do globální proměnné bMcsAxesReady, jejíž hodnota se vyčítá v objektu Main.

#### **4.6.4 ACS\_Axes\_ActualPosition**

ACS\_Axes\_ActualPosition vyčítá pomocí funkčních bloků MC\_ReadActualPosition aktuální pozici jednotlivých fyzických os. První, druhá a čtvrtá osa je ve stupních, třetí osa je v milimetrech. Aktuální pozice se ukládá do lokálních proměnných nPos1-4. Funkční blok MC\_ReadActualPosition je detailněji popsán v kapitole 4.7.

#### **4.6.5 Kinematika\_transformace**

Kinematika\_transformace obsahuje kód, který slouží ke konverzi mezi souřadnicovým systémem os (ACS) a kartézským souřadnicovým systémem. Skládá se z funkčních bloků, funkcí, struktur a výčtového typu enum. Jednotlivé prvky jsou detailněji popsány v kapitole 4.7. Na začátku si objekt vyčte identifikační čísla všech os a ověří si jejich stav. Poté podle toho začne provádět jednotlivé operace.

## 4.7 Objekty použité v kinematické transformaci

Písmena před podtržítkem v názvu jednotlivých objektů určují, o jaký typ objektu se jedná.

### 4.7.1 MC\_ReadActualPosition

Funkční blok slouží k vyčtení aktuální pozice osy. Příklad funkčního bloku pro vyčtení aktuální pozice první osy je zobrazen na výpisu 4.7. [34]

Výpis 4.7: Funkční blok MC\_ReadActualPosition

```
1 fbPos1(  
2   Axis:= gvl1.io_M1,  
3   Enable:= TRUE,  
4   Valid=> ,  
5   Busy=> ,  
6   Error=> ,  
7   ErrorID=> ,  
8   Position=> nPos1);
```

Popis parametrů funkčního bloku: [34]

- Axis - výběr osy
- Enable - spuštění funkčního bloku
- Valid - nastaven na true, pokud má získaná pozice validní hodnotu
- Busy - indikuje, že je funkční blok aktivní
- Error - nastaví se na true, pokud se vyskytne nějaká chyba
- ErrorID - pokud se vyskytne chyba, vypíše její identifikační číslo
- Position - vypíše aktuální pozici osy

### 4.7.2 ST\_KinAxes

Jedná se o strukturu definující osy, které tvoří kinematický systém. Struktura se skládá ze dvou polí. První pole obsahuje identifikační čísla os, které tvoří MCS, druhé pole obsahuje identifikační čísla os, které tvoří ACS. Příklad zápisu do struktury pro první osu (ACS) a osu X (MCS) je zobrazen na výpisu 4.8. [33]

Výpis 4.8: Struktura ST\_KinAxes

```
1 VAR  
2   stAxesConfig : ST_KinAxes;  
3   io_M1        : AXIS_REF; (*v projektu je tento  
4                               řádek umístěn v GVL1*)
```

```

5   io_X          : AXIS_REF; (*v projektu je tento
6                               řádek umístěn v GVL1*)
7   END_VAR
8
9   stAxesConfig.nAxisIdsAcs[1] := gvl1.io_M1.NcToPlc.AxisId;
10  stAxesConfig.nAxisIdsMcs[1] := gvl1.io_X.NcToPlc.AxisId;

```

Struktura stAxesConfig je typu ST\_KinAxes. AXIS\_REF je datový typ, který obsahuje informace o ose a slouží jako rozhraní propojující PLC s NC. Poslední dva řádky slouží k vyčtení identifikačních čísel jednotlivých os a následnému uložení do daných polí s identifikačními čísly. [33, 35]

### 4.7.3 FB\_KinConfigGroup

Funkční blok konfiguruje osy vzhledem ke kinematické transformaci. Vyčte ACS a MCS osy definované v položce stAxesList a nastaví je v kinematické skupině stKinRefIn. Aby proběhl funkční blok v pořádku, je potřeba softwarově povolit všechny ACS osy pomocí bloku MC\_Power, toto zajišťuje objekt Axes\_control\_ACS popsáný v kapitole 4.6.2. Použitý funkční blok z projektu je zobrazen na výpisu 4.9. [33]

Výpis 4.9: Funkční blok FB\_KinConfigGroup

```

1  fbKinConfigGroup(
2    bExecute:= gvl1.bExecuteKinConfGroup,
3    bCartesianMode:= gvl1.bCartesianMode,
4    stAxesList:= stAxesConfig,
5    stKinRefIn:= in_stKinToPlc,
6    bBusy=> ,
7    bDone=> ,
8    bError=> ,
9    nErrorId=> );

```

Popis parametrů funkčního bloku: [33]

- bExecute - spuštění funkčního bloku na náběžnou hranu
- bCartesianMode - pokud je ve stavu false, tak lze pohybovat přímo s osami ACS; pokud je ve stavu true, tak je nastavena transformace z MCS na ACS a s ACS osami nelze hýbat přímo
- stAxesList - určuje ACS a MCS osy obsažené v konfiguraci
- stKinRefIn - určuje kinematickou skupinu konfigurace



- bBusy - výstup je nastaven na true po spuštění bloku pomocí bExecute, zůstává nastavený na true po celou dobu běhu funkčního bloku; po dobu, kdy je ve stavu true, nelze zadávat nové vstupní hodnoty
- bDone - nastaví se na true po úspěšném ukončení operace
- bError - nastaví se na true, pokud se během vykonávání funkčního bloku objeví chyba
- nErrorId - obsahuje identifikační číslo chyby posledního spuštěného příkazu

#### 4.7.4 FB\_KinResetGroup

Funkční blok, který resetuje kinematickou skupinu a všechny ACS a MCS osy. Pokud jsou všechny osy povoleny a skupina byla v kartézském souřadnicovém systému, tak se skupina vrátí do stavu KinStatus\_Ready. Pokud skupina nebyla v kartézském souřadnicovém systému nebo osy nejsou povolené, tak bude skupina ve stavu KinStatus\_Empty. Tyto stavy jsou popsány v kapitole 4.7.5. Funkční blok je zobrazen na výpisu 4.10. [33]

Výpis 4.10: Funkční blok FB\_KinResetGroup

```

1 fbKinResetGroup (
2     bExecute := gvl1.bResetKinGroup ,
3     nItpChannelId := 2 ,
4     stKinRefIn := in_stKinToPlc ,
5     stAxesList := stAxesConfig ,
6     bBusy => ,
7     bDone => ,
8     bError => ,
9     nErrorId => );

```

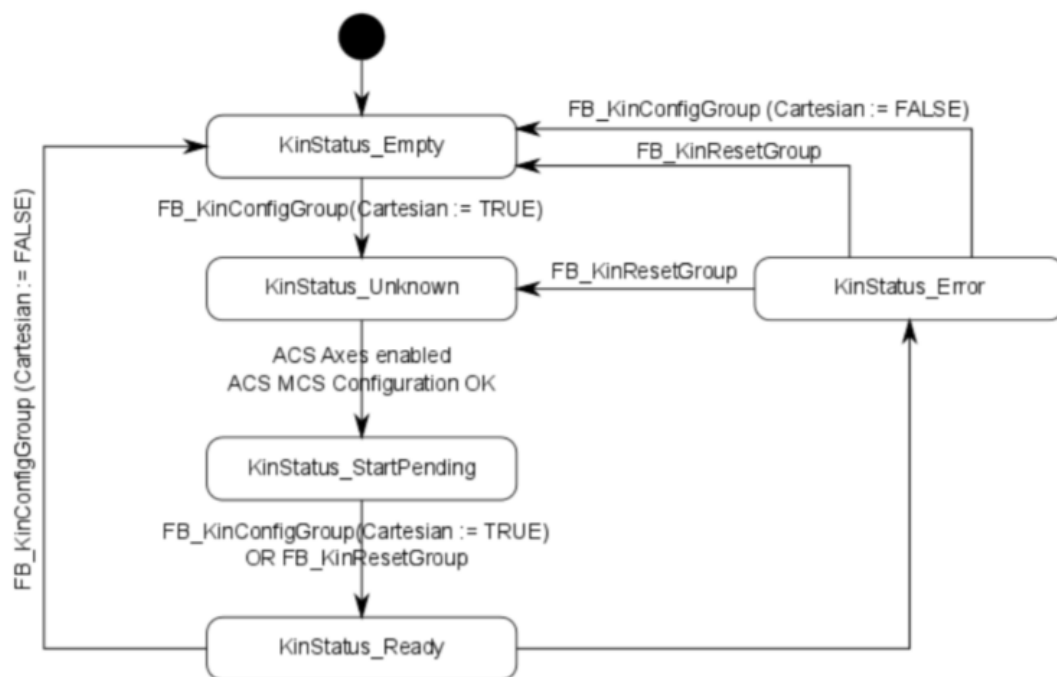
Popis parametrů funkčního bloku: [33]

- bExecute - spuštění funkčního bloku na náběžnou hranu
- nItpChannelId - identifikační číslo interpolačního kanálu; pokud vstup není roven nule, tak se interpolační kanál resetuje
- stKinRefIn - určuje kinematickou skupinu konfigurace
- stAxesList - určuje ACS a MCS osy obsažené v konfiguraci
- bBusy - výstup je nastaven na true po spuštění bloku pomocí bExecute a zůstává nastavený na true po celou dobu běhu funkčního bloku; po dobu, kdy je ve stavu true, nelze zadávat nové vstupní hodnoty
- bDone - výstup se nastaví na true, pokud byl funkční blok vykonán úspěšně
- bError - výstup se nastaví na true, pokud během vykonávání funkčního bloku nastane chyba

- nErrorId - obsahuje identifikační číslo chyby posledního spuštěného bloku

#### 4.7.5 E\_KinStatus

Jedná se o výčtový typ enum, který definuje stav kinematické skupiny. Na obrázku 4.3 je zobrazen vývojový diagram popisující stavy kinematické skupiny pomocí tohoto výčtového typu. [33]



Obr. 4.3: Vývojový diagram pro E\_KinStatus [33]

Důležité jsou tyto stavy: [33]

- KinStatus\_Empty - lze pohybovat s osami ACS, není povolena žádná transformace
- KinStatus\_Ready - lze pohybovat s osami MCS, transformace je aktivní
- KinStatus\_InvalidItfVersion - verze rozhraní těchto cyklických kanálů nepodporuje funkci nebo funkční blok

Aby bylo možné dosáhnout stavu KinStatus\_Ready, je potřeba softwarově povolit všechny ACS osy pomocí funkčního bloku MC\_Power, který je popsán v kapitole 4.4.1. [33]

### 4.7.6 F\_KinGetChnOperationState

Tato funkce cyklicky vyčítá aktuální stav kinematické skupiny. Zápis funkce v projektu je zobrazen na výpisu 4.11. [33]

Výpis 4.11: Funkce F\_KinGetChnOperationState

```
1 VAR
2   stKinRefIn      AT %I*   : NCTOPLC_NCICHANNEL_REF;
3   nErrId          : UDINT;
4   eKinOperationState : E_KinStatus;
5 END_VAR
6
7 IF F_KinGetChnOperationState(stKinRefIn) <>
   KinStatus_InvalidItfVersion THEN
8   eKinOperationState :=
       F_KinGetChnOperationState(stKinRefIn);
9 ELSE
10  nErrId := F_KinGetChnOperationState(stKinRefIn);
11 END_IF
```

Parametr stKinRefIn určuje kinematickou skupinu konfigurace. Návrátová hodnota funkce je E\_KinStatus, tato hodnota je popsána v kapitole 4.7.5. [33]

### 4.7.7 F\_KinGetAcsMcsAxisIds

Tato funkce čte nastavené ACS a MCS osy cyklického rozhraní a zapisuje jejich identifikační čísla do položky stAxesList. Zápis funkce v projektu je zobrazen na výpisu 4.12. [33]

Výpis 4.12: Funkce F\_KinGetAcsMcsAxisIds

```
1 VAR
2   stAxesList      : ST_KinAxes;
3   stKinRefIn      AT %I*   : NCTOPLC_NCICHANNEL_REF;
4   nErrId2         : UDINT;
5 END_VAR
6
7 nErrId2 := F_KinGetAcsMcsAxisIds(stAxesList, stKinRefIn);
8 IF nErrId2 = 0 THEN
9   ; //seznam os je v poradku
10 END_IF
```

Popis zápisu funkce: [33]

- stAxesList - obsahuje identifikační čísla ACS a MCS os
- stKinRefIn - přístupné pouze pro čtení a jedná se o strukturu cyklického kanálového rozhraní mezi kinematickým kanálem a PLC
- nErrId2 - proměnná obsahující kód chyby

#### 4.7.8 FB\_KinCalcTrafo

Funkční blok, který počítá dopřednou a inverzní kinematiku. Dokáže vypočítat kinematiku i bez vytvoření kinematické skupiny pomocí funkčního bloku

FB\_KinConfigGroup. Manipulátoru se zadají požadované pozice v kartézském souřadnicovém systému, mezi kterými se má pohybovat, a tento funkční blok je přepočte na souřadnicový systém os. Zápis funkčního bloku použitého v projektu je zobrazen na výpisu 4.13. [33]

Výpis 4.13: Funkční blok FB\_KinCalcTrafo

```
1 gvl1.uScaraMetaInfoIn.eScara :=  
    E_KinMetaInfoScara.scaraLeftArm;  
2  
3 fbKinCalcTrafo(  
4     bExecute:= gvl1.bExecuteKinCalc,  
5     bForward:= gvl1.bKinTypeBwdTrafo,  
6     oidTrafo:= gvl1.oidTrafo,  
7     stAxesPosIn:= gvl1.stAxesPosIn,  
8     stAxesPosOut:= gvl1.stAxesPosOut,  
9     uMetaInfoIn:= gvl1.uScaraMetaInfoIn,  
10    uMetaInfoOut:= gvl1.uScaraMetaInfoOut,  
11    bBusy=> ,  
12    bDone=> ,  
13    bError=> ,  
14    nErrorId=> );
```

Popis parametrů funkčního bloku: [33]

- bExecute - spuštění funkčního bloku na náběžnou hranu
- bForward - určuje, zda se bude počítat dopředná nebo inverzní kinematika
- oidTrafo - Object-ID (identifikační číslo) objektu, pro který se bude počítat kinematika, v tomto případě se jedná o identifikační číslo SCARA robota
- stAxesPosIn - pole obsahující vstupní pozice transformace; pro dopřednou kinematiku se jedná o pozice kloubů SCARA robota, pro inverzní kinematiku se jedná o kartézské souřadnice os

- stAxesPosOut - pole obsahující výsledné pozice transformace; pro dopřednou kinematiku se jedná o kartézské souřadnice os, pro inverzní kinematiku se jedná o pozice kloubů SCARA robota
- uMetaInfoIn - používá se v případě, kdy k řešení vede více různých možných nastavení robota, lze si pomocí tohoto parametru vybrat preferované řešení
- uMetaInfoOut - používá se v případě, kdy je pro transformaci více možných řešení; definuje nalezené řešení
- bBusy - výstup je nastaven na true po spuštění bloku pomocí bExecute a zůstává nastavený na true po celou dobu běhu funkčního bloku; po dobu, kdy je ve stavu true, nelze zadávat nové vstupní hodnoty
- bDone - nastaví se na true po úspěšném ukončení operace
- bError - nastaví se na true, pokud se během vykonávání funkčního bloku objeví chyba
- nErrorId - obsahuje identifikační číslo chyby posledního spuštěného příkazu

#### 4.7.9 FB\_KinUnlockTrafoParam

Funkční blok odemyká parametry kinematické skupiny pro zápis. Po odemknutí parametrů lze změnit vybraný prvek pomocí bloku ADSWRITE. V projektu byl použit pro odemknutí a změnu parametru Tool offset OID, který udává identifikační číslo pracovní hlavice. Část zdrojového kódu obsahující tuto změnu parametru je zobrazena na výpisu 4.14. [33]

Výpis 4.14: Odemknutí a změna parametru kinematické skupiny

```

1 fbKinUnlockTrafoPar(
2   bExecute:= gvl1.bExecuteUnlock,
3   oidTrafo:= gvl1.oidTrafo,
4   bBusy=> ,
5   bDone=> ,
6   bError=> ,
7   nErrorId=> );
8
9 IF gvl1.bExecuteUnlock THEN
10  fbADSWRITE(
11    NETID:= ' ',
12    PORT:= AMSPORT_RO_NCSAF,
13    IDXGRP:= gvl1.oidTrafo,
14    IDXOFFS:= gvl1.pidTrafo,
15    LEN:= sizeof(gvl1.fParamValue),
16    SRCADDR:= adr(gvl1.fParamValue),

```

```

17     WRITE:= gvl1.bExecuteWritePar ,
18     TMOUT:= ,
19     BUSY=> ,
20     ERR=> ,
21     ERRID=> );

```

Popis parametrů funkčního bloku FB\_KinUnlockTrafoParam: [33]

- bExecute - spuštění příkazu na náběžnou hranu
- oidTrafo - identifikační číslo objektu kinematické transformace (identifikační číslo SCARA robota)
- bBusy - výstup je nastaven na true po spuštění bloku pomocí bExecute a zůstává nastavený na true po celou dobu běhu funkčního bloku; po dobu, kdy je ve stavu true, nelze zadávat nové vstupní hodnoty
- bDone - nastaví se na true po úspěšném ukončení operace
- bError - nastaví se na true, pokud se během vykonávání funkčního bloku objeví chyba
- nErrorId - obsahuje identifikační číslo chyby posledního spuštěného příkazu

Pro blok ADSWRITE jsou důležité tyto parametry: [36]

- IDXGRP - identifikační číslo objektu kinematické transformace (identifikační číslo SCARA robota)
- IDXOFFS - identifikační číslo parametru Tool offset OID
- LEN - velikost čtených dat v bytech
- SRCADDR - adresa vyrovnávací paměti, ze které mají být načtena data, která se mají zapsat
- WRITE - spuštění zápisu do parametru na náběžnou hranu

#### 4.7.10 FB\_KinLockTrafoParam

Po editaci parametrů kinematické skupiny je potřeba skupinu znovu uzamknout a zablokovat tak další zápis. K tomu slouží tento funkční blok. Na výpisu 4.15 je zobrazena část kódu z projektu obsahující tento blok, tato část přímo navazuje na výpis 4.14. [33]

Výpis 4.15: Uzamčení parametrů kinematické skupiny

```

1  fbKinLockTrafoPar(
2      bExecute:= TRUE,
3      oidTrafo:= gvl1.oidTrafo,
4      bBusy=> ,
5      bDone=> ,
6      bError=> ,

```

```

7      nErrorId=> );
8
9      gv11.bExecuteUnlock := FALSE;
10 END_IF;

```

Popis parametrů funkčního bloku FB\_KinLockTrafoParam: [33]

- bExecute - spuštění příkazu na náběžnou hranu
- oidTrafo - identifikační číslo objektu kinematické transformace (identifikační číslo SCARA robota)
- bBusy - výstup je nastaven na true po spuštění bloku pomocí bExecute, zůstává nastavený na true po celou dobu běhu funkčního bloku; po dobu, kdy je ve stavu true, nelze zadávat nové vstupní hodnoty
- bDone - nastaví se na true po úspěšném ukončení operace
- bError - nastaví se na true, pokud se během vykonávání funkčního bloku objeví chyba
- nErrorId - obsahuje identifikační číslo chyby posledního spuštěného příkazu

# Závěr

Na začátku této práce jsem popsal typy robotických systémů. Robotické systémy jsem rozdělil na tři základní typy. Zaměřil jsem se hlavně na manipulační zařízení, jelikož do této skupiny spadá i robot typu SCARA. Manipulační zařízení jsem popsal z hlediska jejich vývoje a použití.

Následně jsem se věnoval stavbě robota z hlediska kinematiky. Popsal jsem základní typy kinematických dvojic, které jsou nutné pro správné zkonstruování robota. Poté jsem popsal základní typy kinematických struktur, které vznikly z různých spojení těchto dvojic. Tyto struktury tvoří základní podobu zkonstruovaného robota, z nich se vyvíjely další typy robotů. Z jednoho z těchto typů struktur se vyvinul i robot SCARA, jenž byl hlavním tématem této práce.

Tomuto typu jsem věnoval největší část práce. Začal jsem od popisu obecných informací, jako je například jeho vznik v roce 1978, jak byl zkonstruován a k čemu vlastně tento robot slouží, abych se díky tomu poté mohl lépe seznámit s manipulátorem H554BN od firmy Epson. U manipulátoru H554BN jsem se věnoval jeho konstrukci a parametrům, jako je například jeho dosah, nebo pohon.

K fungování manipulátoru je nutný řídicí systém. Ten byl použit od firmy Beckhoff. Popsal jsem použité servoměniče včetně jejich typu řízení a připojené vstupní a výstupní terminály, které se používají pro senzory. Samostatné kapitoly jsem věnoval průmyslovým počítačům, které toto vše řídí a komunikačnímu rozhraní EtherCAT, jenž propojuje všechna tato zařízení.

Aby bylo možné manipulátor začít programovat, tak bylo nejdříve nutné zapojit jednotlivé enkodéry. Při tomto zapojování se vyskytly problémy. Zjistil jsem, že deska v základně robota je vyzkratovaná a propojovací kabel nefunkční. Proto bylo nutné nejdříve vymyslet nové řešení, jak nefunkčního robota opravit a zapojit k řídicímu systému, a následně tuto opravu realizovat. Během realizace se několikrát ukázalo, že navržená oprava není vhodná, proto byl navržen jiný způsob opravy. V rámci práce jsem měl za úkol vytvořit kompletní schéma zapojení, to jsem udělal pomocí softwaru WSCAD, se kterým jsem se díky tomu naučil pracovat. Vytvořil jsem dvě schémata. V prvním schématu je kromě propojovacího kabelu mezi manipulátorem a servoměniči kompletní zapojení. Ve druhém schématu se nachází pouze nakreslené propojení manipulátoru se servoměniči podle toho, jak byla realizovaná oprava. Zbýlé zapojení je identické, proto jsem ho do druhého schématu již nezakresloval.

Ve firmě Beckhoff jsem v prosinci 2019 absolvoval čtyřdenní školení, kde se probíral hardware, který firma nabízí. Dále jsem se na školení učil základy s jejich softwarem TwinCAT 3 a základy programování, včetně základů v oblasti nastavení a řízení motorů. Část školení byla věnována bezpečnosti. Na školení jsem se naučil základy programování v jejich softwaru, které jsem poté využil pro naprogramo-



vání manipulátoru H554BN. V této době jsem zároveň vytvářel základní funkce pro samostatný pohyb jednotlivých os, ale neměl jsem ještě kompletně opravený manipulátor, proto jsem si tyto funkce nejdříve otestoval na hardwaru ve firmě Beckhoff, včetně vytvoření celé konfigurace a až poté, co jsem dokončil opravu, jsem implementoval vše přímo na tento manipulátor. Vytvořil jsem také kompletní kinematickou transformaci, aby bylo možné řídit všechny osy jako celek.

Během programování jsem se setkal s problémem, kdy se někdy objevilo rušení na druhé ose manipulátoru, kvůli čemuž následně do servoměniče přišla z enkodéru pro tuto osu špatná data a celý manipulátor se zastavil. Na tuto chybu mohlo mít vliv přerušené stínění uvnitř manipulátoru nebo chybějící stínění na konektoru pro enkodér, který byl připojen do servoměniče. Od mého vedoucího jsem obdržel návrh regulátorů a parametry pro jednotlivé motory. Při spuštění pohybu se manipulátor ne vždy pohyboval správně. Při diagnostice tohoto problému jsem zjistil, že chyba může být buď v špatném nastavení regulátorů, které jsem obdržel, nebo ve špatném nastavení komutačního offsetu servomotorů v měniči. I přes pomoc několika vyučujících a opětovný návrh regulátorů a změnu nastavení komutačních offsetů se mi nepovedlo, aby se manipulátor při všech spuštěních pohyboval správně, ale k těmto chybám stále náhodně dochází. I přes tyto uvedené problémy a předchozí problémy s elektronickým zapojením se mi povedlo mé řešení úspěšně otestovat. Toto řešení je i přes uvedené nedostatky robota funkční a splnil jsem tak všechny body zadání.

# Literatura

- [1] RobotWorx: *Three types of robotic systems* [online]. [cit. 2019-11-11]. Dostupné z: <https://www.robots.com/articles/three-types-of-robotic-systems>
- [2] OPLATEK, František, Miloš LUNER, Jiří OSOBA, Karel SVOBODA a Ladislav ŠMEJKAL. *Automatizace a automatizační technika*. Praha: Computer Press, 2000. Všechny cesty k informacím. ISBN 80-7226-249-1.
- [3] SKOPALÍK, Slavomír. *Sběr dat z výroby z pohledu systémového integrátora. Automa* [online]. 2004, 2004(6) [cit. 2019-11-18]. Dostupné z: [https://automa.cz/cz/casopis-clanky/sber-dat-z-vyroby-z-pohledu-systemoveho-integratora-2004\\_06\\_32367\\_802/](https://automa.cz/cz/casopis-clanky/sber-dat-z-vyroby-z-pohledu-systemoveho-integratora-2004_06_32367_802/)
- [4] ŠTĚPÁN, Petr. *Mobilní robotika: Úvod do mobilní robotiky* [online]. Praha, 2010-09-21 [cit. 2019-12-10]. Dostupné z: [https://cw.fel.cvut.cz/old/\\_media/mkr/lessons/lesson1.pdf](https://cw.fel.cvut.cz/old/_media/mkr/lessons/lesson1.pdf). České vysoké učení technické v Praze, Katedra kybernetiky.
- [5] WICKS, Matt, HOSKE, Mark T., ed. *Robotické řízení - specializované, nebo na bázi PLC? Control Engineering* [online]. 25.04.14 [cit. 2020-06-05]. Dostupné z: <http://www.controlengcesko.com/hlavni-menu/artikuly/artikul/article/roboticke-rizeni-specializovane-nebo-na-bazi-plc/>
- [6] KOHOUT, Luděk. *Roboty a manipulátory* [online]. Kutná Hora [cit. 2019-12-23]. Dostupné z: [http://edumat.cz/texty/Roboty\\_manipulatory.pdf](http://edumat.cz/texty/Roboty_manipulatory.pdf). VOŠ a SPŠ Kutná Hora.
- [7] BENEŠL, Tomáš. *Optimální řízení průmyslových robotických systémů*. Brno, 2019. Pojednání o dizertační práci. Vysoké učení technické v Brně.
- [8] HAVLÍČEK, Daniel. *Co jsou to SCARA roboty?* [online]. 2019-03-23 [cit. 2019-12-26]. Dostupné z: <https://factoryautomation.cz/co-jsou-to-scara-roboty/>
- [9] Robot Hall of Fame: *SCARA* [online]. [cit. 2019-12-26]. Dostupné z: <http://www.robothalloffame.org/inductees/06inductees/scara.html>
- [10] Flexibowl: *SCARA Robot* [online]. [cit. 2019-12-26]. Dostupné z: <https://www.flexibowl.com/scara-robot.html>
- [11] SEIKO EPSON CORPORATION. *BN TYPE: SCARA ROBOT*. Rev. 5.

- [12] YASKAWA ELECTRIC CORPORATION. *Series SGM/SGMP/SGDA: User's manual*. TSE-S800-15C. Japan, 2002-07.
- [13] BECKHOFF. *Startup: Servo Drive AX5000* [online]. Version: 5.9. 2018-05-24 [cit. 2019-12-28]. Dostupné z: [https://download.beckhoff.com/download/document/motion/ax5000\\_startup\\_en.pdf](https://download.beckhoff.com/download/document/motion/ax5000_startup_en.pdf)
- [14] ElektroPrůmysl.cz: *Typy řízení měniče* [online]. 9. Září 2014 [cit. 2020-06-03]. ISSN 2571-0761. Dostupné z: <https://www.elektroprumysl.cz/energetika/typy-řízení-menice>
- [15] ETHERCAT TECHNOLOGY GROUP. *EtherCAT - The Ethernet Fieldbus* [online]. 2018 [cit. 2020-01-01]. Dostupné z: [https://www.ethercat.org/download/documents/ETG\\_Brochure\\_EN.pdf](https://www.ethercat.org/download/documents/ETG_Brochure_EN.pdf)
- [16] BECKHOFF. *EK110x, EK15xx: EtherCAT Bus Coupler* [online]. Version 3.7. 2018-11-28 [cit. 2020-01-01]. Dostupné z: [https://download.beckhoff.com/download/document/io/ethercat-terminals/ek110x\\_ek15xxen.pdf](https://download.beckhoff.com/download/document/io/ethercat-terminals/ek110x_ek15xxen.pdf)
- [17] BECKHOFF. *EL2008* [online]. c2019 [cit. 2020-01-01]. Dostupné z: <https://www.beckhoff.com/EL2008/>
- [18] BECKHOFF. *EL1124* [online]. c2018 [cit. 2020-01-01]. Dostupné z: <https://www.beckhoff.com/EL1124/>
- [19] BECKHOFF. *EL9xxx: EtherCAT System Terminals* [online]. Version 3.9. 2019-05-10 [cit. 2020-01-01]. Dostupné z: <https://download.beckhoff.com/download/document/io/ethercat-terminals/el9xxxen.pdf>
- [20] BECKHOFF. *PC-based control technology: Průmyslová PC* [online]. [cit. 2020-01-02]. Dostupné z: <https://www.beckhoff.cz/cz/default.htm?beckhoff/products.htm>
- [21] BECKHOFF. *C6015-0010: Ultra compact Industrial PC* [online]. Version 1.3. 2018-04-12 [cit. 2020-01-02]. Dostupné z: <https://infosys.beckhoff.com/content/1033/ipcinfosys/PDF/C6015-0010.pdf>
- [22] BECKHOFF. *Školení TwinCAT3 PLC: Licencování TwinCAT 3*. Brno, 2018.
- [23] SEIKO EPSON CORPORATION. *BN, BNA, BNA-CL, SRC-320: Maintenance manual*. Rev. 1

- [24] About WSCAD: Solutions. *WSCAD* [online]. [cit. 2020-06-02]. Dostupné z: <https://www.wscad.com/en/wscad-at-a-glance/>
- [25] What are Numerical Control Machine? What are NC Machines? In: *Bright Hub Engineering* [online]. 2009-11-12 [cit. 2020-06-03]. Dostupné z: <https://www.brighthubengineering.com/manufacturing-technology/55670-what-is-numerical-control-machine/>
- [26] BECKHOFF. *BECKHOFF AX5000: Commissioning: Commutation methods.* [online]. [cit. 2020-06-03]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/ax5000\\_\\_usermanual/html/AX5000\\_Kommutierungsverfahren.htm&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/ax5000__usermanual/html/AX5000_Kommutierungsverfahren.htm&id=)
- [27] BECKHOFF. *TwinCAT PLC Control: Project Structure: Project Components* [online]. [cit. 2020-06-03]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-control/html/tcplcctrl\\_setup.htm&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-control/html/tcplcctrl_setup.htm&id=)
- [28] BECKHOFF. *MC\_Power* [online]. [cit. 2020-01-04]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib\\_tc2\\_mc2/36028797089013387.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib_tc2_mc2/36028797089013387.html&id=)
- [29] BECKHOFF. *MC\_Reset* [online]. [cit. 2020-01-04]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib\\_tc2\\_mc2/36028797089014923.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib_tc2_mc2/36028797089014923.html&id=)
- [30] BECKHOFF. *MC\_Stop* [online]. [cit. 2020-01-04]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib\\_tc2\\_mc2/45035996343813515.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib_tc2_mc2/45035996343813515.html&id=)
- [31] BECKHOFF. *MC\_MoveAbsolute* [online]. [cit. 2020-01-04]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib\\_tc2\\_mc2/36028797089058699.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib_tc2_mc2/36028797089058699.html&id=)
- [32] BECKHOFF. *MC\_MoveRelative* [online]. [cit. 2020-01-04]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib\\_tc2\\_mc2/36028797089060235.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib_tc2_mc2/36028797089060235.html&id=)
- [33] BECKHOFF. *TF5110 - TF5113 / TC3 Kinematic Transformation: manual* [online]. Version 1.3. 2020-04-15 [cit. 2020-05-27]. Dostupné z: [https://download.beckhoff.com/download/document/automation/twincat3/TF5110-TF5113\\_TC3\\_Kinematic\\_Transformation\\_EN.pdf](https://download.beckhoff.com/download/document/automation/twincat3/TF5110-TF5113_TC3_Kinematic_Transformation_EN.pdf)

- [34] BECKHOFF. *MC\_ReadActualPosition* [online]. [cit. 2020-05-30]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib\\_tc2\\_mc2/45035996343760395.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib_tc2_mc2/45035996343760395.html&id=)
- [35] BECKHOFF. *AXIS\_REF* [online]. [cit. 2020-05-31]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib\\_tc2\\_mc2/54043195598578315.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib_tc2_mc2/54043195598578315.html&id=)
- [36] BECKHOFF. *ADSWRITE* [online]. [cit. 2020-06-02]. Dostupné z: [https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib\\_tc2\\_system/45035996304573067.html&id=](https://infosys.beckhoff.com/english.php?content=../content/1033/tcplc-lib_tc2_system/45035996304573067.html&id=)

# Seznam symbolů, veličin a zkratk

<b>AC</b>	Alternating Current – střídavý proud
<b>ACS</b>	Axis Coordinate System – souřadnicový systém osy
<b>CAT 5E</b>	Category 5 Enhanced – vylepšená 5. kategorie
<b>DC</b>	Direct Current - stejnosměrný proud
<b>DVI</b>	Digital Visual Interface – rozhraní pro propojení počítače a zobrazovacího zařízení
<b>D-SUB CAN 50</b>	D-subminiature CANON– druh konektoru CANON
<b>EtherCAT</b>	Ethernet for Control Automation Technology – ethernet pro řídicí systémy
<b>FB</b>	Function Block – funkční blok
<b>FGND</b>	Frame Ground – stínění zařízení
<b>GVL</b>	Global Variable List – seznam globálních proměnných
<b>ID</b>	Identification – identifikace
<b>IoT</b>	Internet of Things – internet věcí
<b>I/O</b>	Input/Output – vstup/výstup
<b>LED</b>	Light Emitting Diode – světlo emitující dioda
<b>MC</b>	Motion Control – řízení pohybu
<b>MCS</b>	Machine Coordinate System – souřadnicový systém stroje
<b>NC</b>	Numerical Control – numerické řízení
<b>OID</b>	Object Identification – identifikační číslo objektu
<b>PLC</b>	Programmable Logic Controller – programovatelný logický automat
<b>POU</b>	Program Organization Unit – organizační jednotka programu
<b>RAM</b>	Random Access Memory – paměť s náhodným přístupem
<b>RJ</b>	Registered Jack – typ konektoru
<b>SCARA</b>	Selective Compliant Articulated Robot Arm nebo Selective Compliant Assembly Robot Arm – selektivní kompatibilní kloubové robotické rameno
<b>SSD</b>	Solid State Drive
<b>Tc2</b>	TwinCAT 2 – starší vývojové prostředí firmy Beckhoff
<b>TwinCAT</b>	The Windows Control Automation Technology – vývojové prostředí firmy Beckhoff
<b>UCS</b>	User Coordinate System – uživatelský souřadnicový systém
<b>USB</b>	Universal Serial Bus – univerzální sériová sběrnice
<b>UTP</b>	Unshielded Twisted Pair – nestíněná kroucená dvojlinka
<b>WCS</b>	World Coordinate System – světový souřadnicový systém
$\Phi[^\circ]$	Úhel natočení

$\theta[^\circ]$	Úhel natočení
$\omega[\mathbf{rad} \cdot \mathbf{s}^{-1}]$	Úhlová rychlost (v projektu $[^\circ/\mathbf{s}]$ )
$I[\mathbf{A}]$	Proud
$I_N[\mathbf{A}]$	Jmenovitý proud
$l[\mathbf{m}]$	Délka
$M_N[\mathbf{N} \cdot \mathbf{m}]$	Jmenovitý točivý moment
$m[\mathbf{kg}]$	Hmotnost
$n_N[\mathbf{ot/min}]$	Jmenovité otáčky
$P_N[\mathbf{W}]$	Jmenovitý výkon
$p[\mathbf{Pa}]$	Tlak
$R[\Omega]$	Elektrický odpor
$t[^\circ\mathbf{C}]$	Teplota
$U[\mathbf{V}]$	Napětí
$-[\mathbf{B}]$	Jednotka informace
$-[\mathbf{mm}/^\circ]$	Poměr mezi třetí a čtvrtou osou

# Seznam příloh

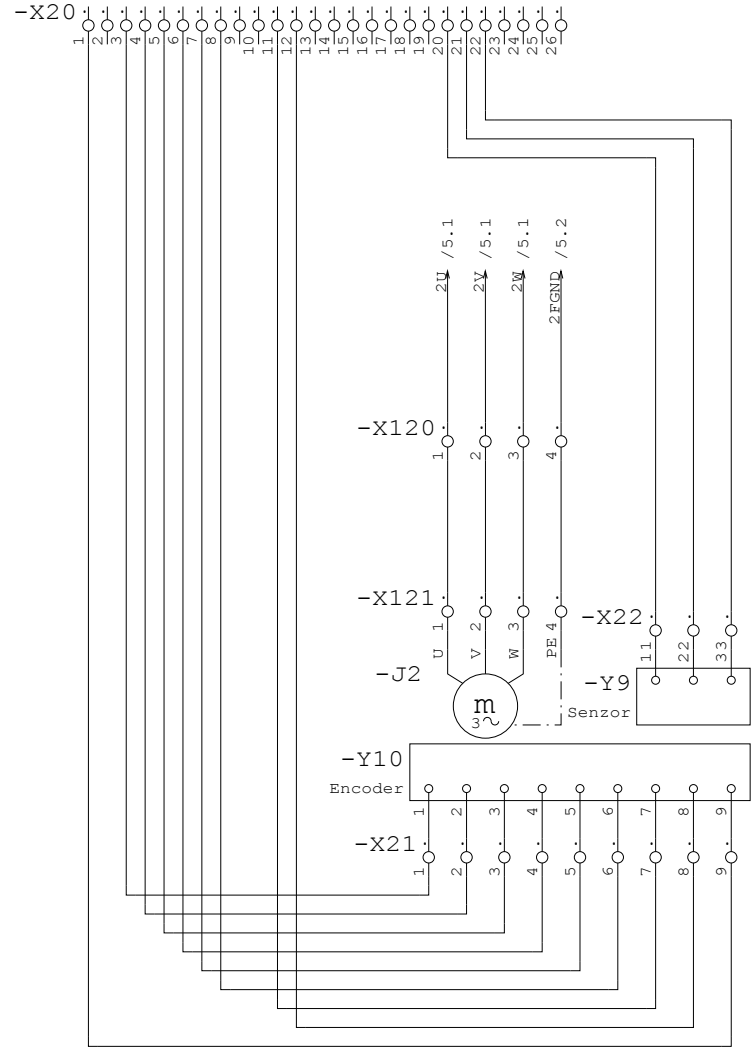
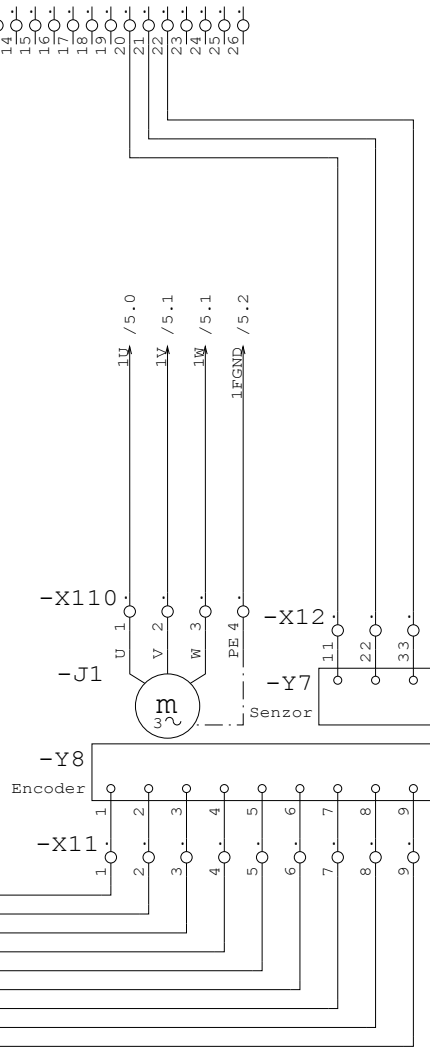
A Schéma zapojení před provedenou opravou	60
B Schéma zapojení s propojovacím kabelem	67
C Obsah přiloženého CD	73



## **A Schéma zapojení před provedenou opravou**

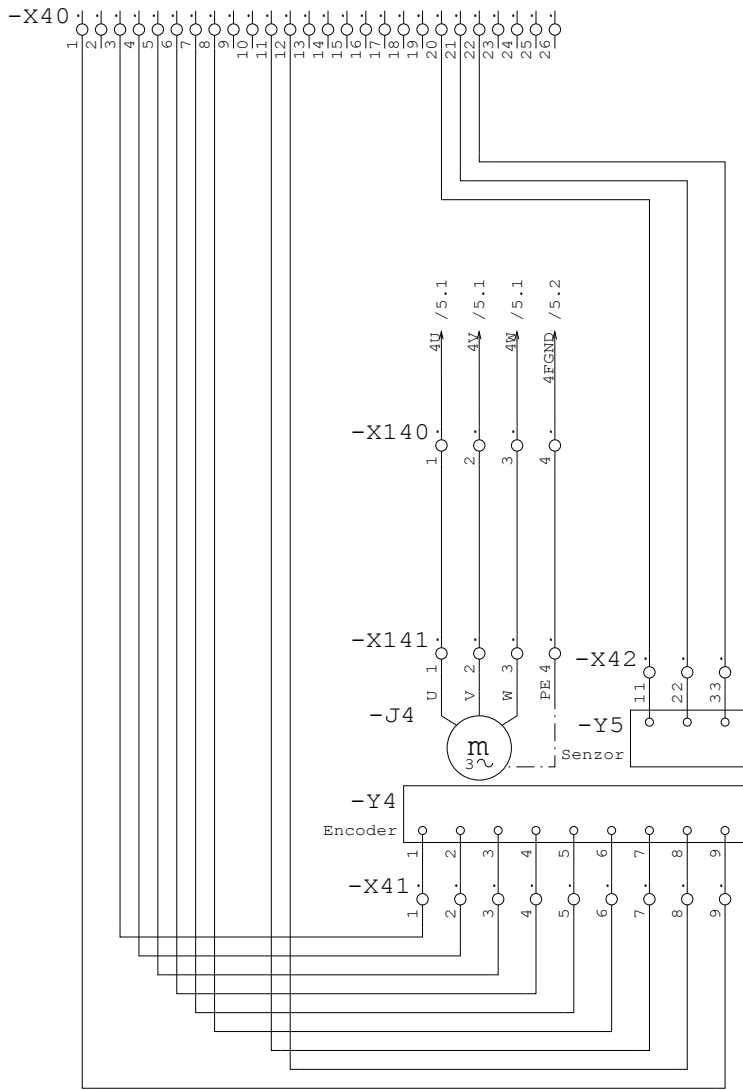
EDUCATION

EDUCATION



				Date	14.11.19			Project number	Unit	=	
				Work					Field	+	
				Check							
Status	Rev.	Date	Name	Norm	DIN 81346	Created for	Created by	Drawing number		sheet 1	
										of 6	

F



F

8

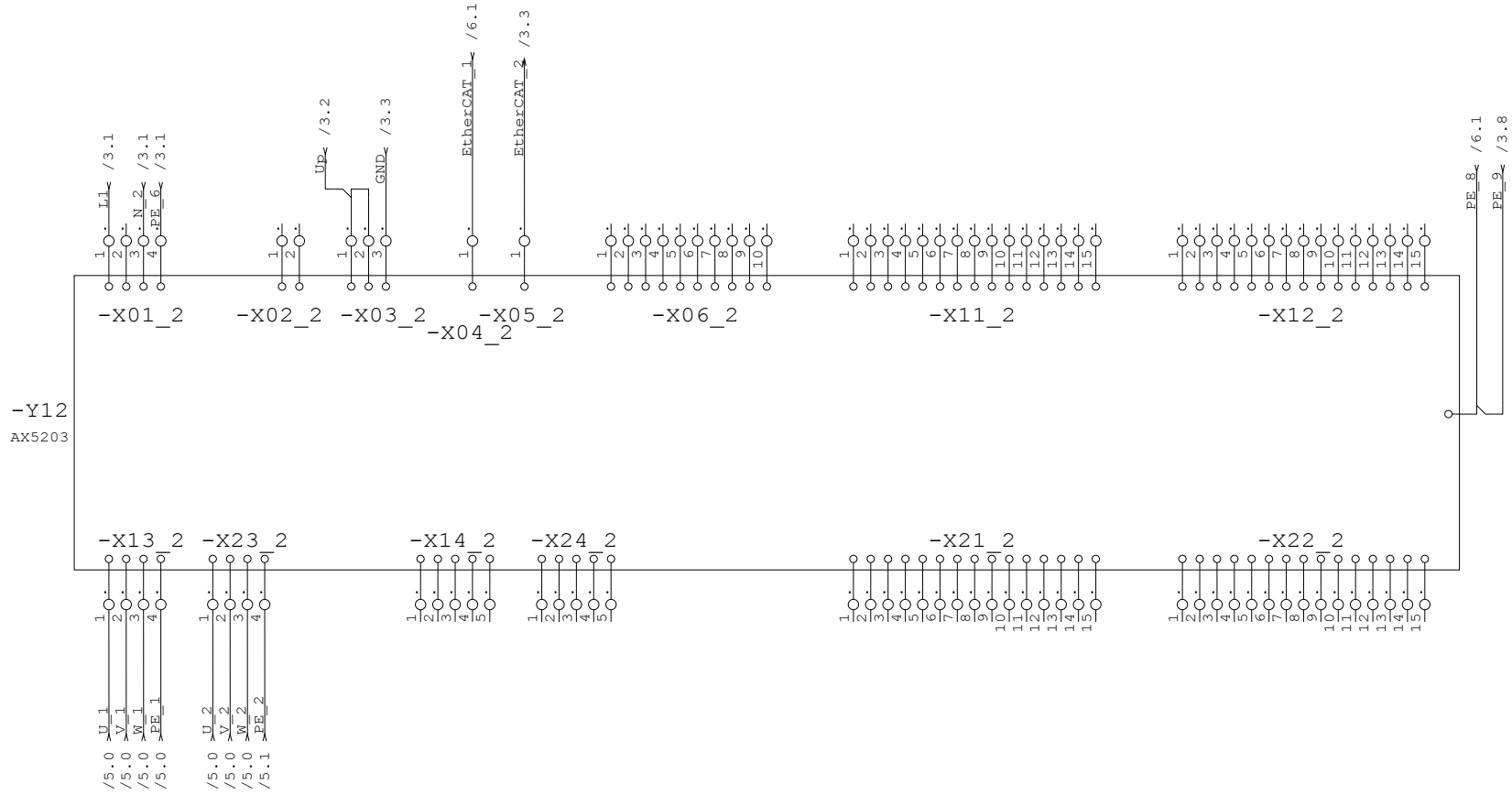
F



0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

EDUCATION

EDUCATION



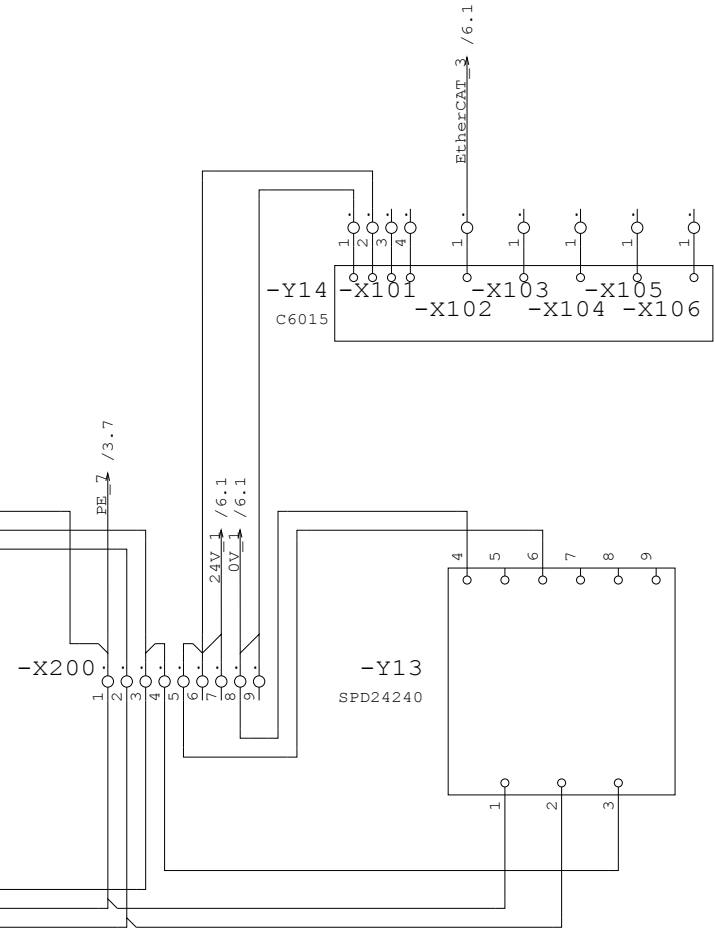
				Date	14.11.19		 ELECTRICAL ENGINEERING		Project number		Unit	=		
				Work						Field	+			
				Check										
Status	Rev.	Date	Name	Norm	DIN 81346	Created for	Created by			Drawing number	sheet		4	
												of		6



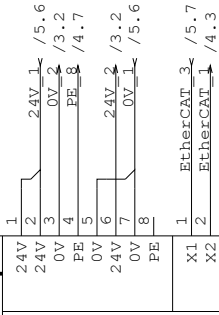
F



# EDUCATION



				Date	14.11.19				Project number		Unit				
				Work					Field		+				
				Check											
Status	Rev.	Date	Name	Norm	DIN 81346	Created for	Created by				Drawing number		sheet	5	
														of	6

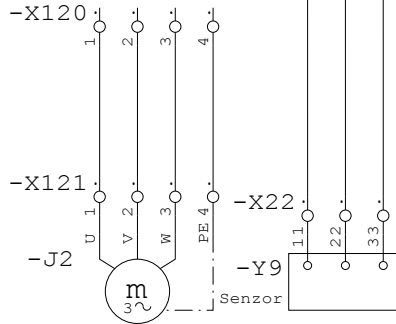
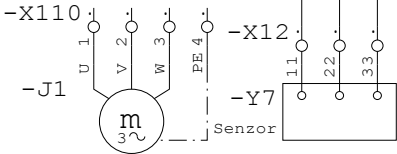


0	1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---	---

## **B Schéma zapojení s propojovacím kabelem**



Fr



F

				Date	06.01.20				Project number		Unit	=				
				Work									Field	+		
				Check												
Status	Rev.	Date	Name	Norm	DIN 81346	Created for	Created by				Drawing number		sheet	1		
													of	5		

Fi



012345678

EDUCATION

/5.5	1A_2	1	O	1A	/1.1
/5.5	1A_NEG_2	2	O	1A_NEG	/1.1
/5.5	1B_2	3	O	1B	/1.1
/5.5	1B_NEG_2	4	O	1B_NEG	/1.1
/5.5	1Z_2	5	O	1Z	/1.1
/5.5	1Z_NEG_2	6	O	1Z_NEG	/1.1
/5.5	5V_ENC1_2	7	O	5V_ENC1	/1.1
/5.4	EGND1_2	8	O	EGND1	/1.1
		9	O	EGND1	/1.1
		10	O	1HOME	/1.3
		11	O	24V_1	/1.3
		12	O	RGND1	/1.3
/5.5	2A_2	13	O	2A	/1.5
/5.5	2A_NEG_2	14	O	2A_NEG	/1.5
/5.5	2B_2	15	O	2B	/1.5
/5.5	2B_NEG_2	16	O	2B_NEG	/1.5
/5.5	2Z_2	17	O	2Z	/1.5
		18	O	FGND3	/2.1
/4.5	EGND3_2	19	O	EGND3	/2.1
/4.5	5V_ENC3_2	20	O	5V_ENC3	/2.1
/4.5	3Z_NEG_2	21	O	3Z_NEG	/2.1
/4.5	3Z_2	22	O	3Z	/2.1
/4.5	3B_NEG_2	23	O	3B_NEG	/2.1
/4.5	3B_2	24	O	3B	/2.1
/4.6	3A_NEG_2	25	O	3A_NEG	/2.0
/4.5	3A_2	26	O	3A	/2.0
		27	O	RGND2	/1.8
		28	O	24V_2	/1.7
		29	O	2HOME	/1.7
		30	O	FGND2	/1.6
/5.4	EGND2_2	31	O	EGND2	/1.6
/5.5	5V_ENC2_2	32	O	5V_ENC2	/1.5
/5.5	2Z_NEG_2	33	O	2Z_NEG	/1.5
		34	O	3HOME	/2.3
		35	O	24V_3	/2.3
		36	O	RGND3	/2.3
		37	O		
		38	O		
/4.5	4A_2	39	O	4A	/2.5
/4.6	4A_NEG_2	40	O	4A_NEG	/2.5
/4.5	4B_2	41	O	4B	/2.5
/4.5	4B_NEG_2	42	O	4B_NEG	/2.5
/4.5	4Z_2	43	O	4Z	/2.6
/4.5	4Z_NEG_2	44	O	4Z_NEG	/2.6
/4.5	5V_ENC4_2	45	O	5V_ENC4	/2.6
/4.5	EGND4_2	46	O	EGND4	/2.6
		47	O	FGND4	/2.6
		48	O	4HOME	/2.8
		49	O	24V_4	/2.8
		50	O	RGND4	/2.8

EDUCATION

				Date	06.01.20	 ELECTRICAL ENGINEERING		Project number		Unit	=		
				Work						Field	+		
				Check							Drawing number		sheet
Status	Rev.	Date	Name	Norm	DIN 81346	Created for	Created by					of	5

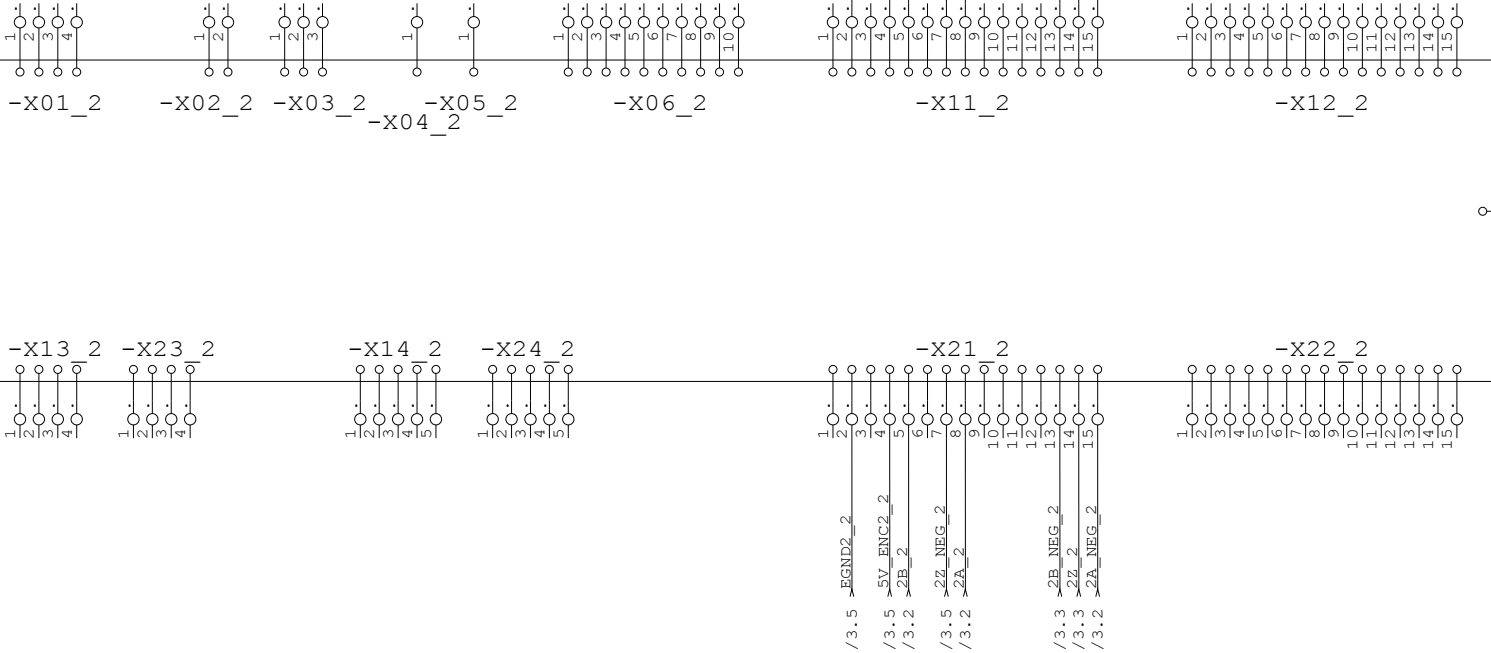
F

F

EDUCATION

EDUCATION

-Y12  
AX5203



				Date	06.01.20			Project number		Unit	=	
				Work						Field	+	
				Check						Drawing number		sheet 5 of 5
Status	Rev.	Date	Name	Norm	DIN 81346	Created for	Created by					



## C Obsah přiloženého CD

```
/ ..... kořenový adresář přiloženého CD
├── xmadzi00_BP_SCARA_program.zip ..... Program pro řízení manipulátoru
├── xmadzi00_WSCAD_schema_zapojeni_kabel.zip .. Schéma zapojení propojovacího
    kabelu v softwaru WSCAD
└── xmadzi00_WSCAD_schema_zapojeni_zarizeni.zip ..... Schéma zapojení zařízení
    v softwaru WSCAD
```